



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

TVORBA SIMULÁTORU TECHNOLOGICKÉHO PROCESU

TECHNOLOGICAL PROCESS SIMULATOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jan Wójcik

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ondřej Baštán

BRNO 2021

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Jan Wójcik

ID: 211192

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Tvorba simulátoru technologického procesu

POKYNY PRO VYPRACOVÁNÍ:

Práce se zabývá návrhem a výrobou simulátoru technologického procesu pro školící centrum. Toto zadání bylo vytvořeno ve spolupráci se společností ABB.

1. Seznamte se se zadáním zadavatele (společnosti ABB).
2. Definujte požadavky na simulátor.
3. Navrhněte koncepci HW a jeho zapojení.
4. Realizujte zapojení simulátoru.
5. Implementujte řídicí SW simulátoru.
6. Navrhněte a realizujte konfigurační rozhraní simulátoru.
7. Zařízení otestujte a zhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ: Řízení Technologických procesů [online]. [cit. 2012-09-14]. Dostupné z <http://rtp.webzdarma.cz/>

Termín zadání: 8.2.2021

Termín odevzdání: 24.5.2021

Vedoucí práce: Ing. Ondřej Baštán

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato bakalářská práce se zabývá návrhem a tvorbou simulátoru technologického procesu určeného pro školicí centrum. Simulátor je zařízení představující regulovanou soustavu, která je řízena programovatelným logickým automatem. Je navržen tak, že požadované stavy a signály technologického procesu jsou vhodně zobrazovány uživateli prostřednictvím operátorského panelu. Uživateli je také umožněno manuálně ovládat simulátor pomocí vstupních tlačítek. Simulátor je zpětně kompatibilní s předchozími verzemi ABB Oslo Demo Simulátoru.

Klíčová slova

Simulátor, technologický proces, PLC, virtualizace, Arduino

Abstract

This bachelor's thesis deals with design and creation of technological process simulator, which is intended to be used in the training center. The simulator is device, that represents system, which is controlled by programmable logic controller. It is designed to suitably display demanded states and signals of the technological process via its operator panel. It is also possible to control simulator manually using the input buttons. The simulator is also backward compatible with the previous versions of ABB Oslo Demo Simulator.

Keywords

Simulator, technological process, PLC, virtualization, Arduino

BIBLIOGRAFICKÁ CITACE:

WÓJCIK, Jan. *Tvorba simulátoru technologického procesu*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/130221>. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Ondřej Baštán.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	Jan Wójcik
VUT ID studenta:	211192
Typ práce:	Bakalářská práce
Akademický rok:	2020/21
Téma závěrečné práce:	Tvorba simulátoru technologického procesu

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucího závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 24. května 2021

podpis autora

Poděkování

Děkuji svému vedoucímu práce Ing. Ondřeji Baštánovi za cenné náměty a rady na vylepšení práce, které mi poskytl během našich konzultací. Dále děkuji vedoucímu marine oddělení EUOPC Ostrava, Petrovi Podhornému, za příležitost vyvíjet simulátor a skvělé podmínky, které mi pro vývoj zařízení poskytl. V neposlední řadě děkuji Radimovi Břenkovi za důležité rady a nápady ohledně mechanické konstrukce a celkový dohled nad tvorbou simulátoru. Rovněž patří dík i Patriku Škodíkovi za pomoc při testování simulátoru.

V Brně dne: 24. května 2021

podpis autora

OBSAH

1. ÚVOD.....	10
2. ANALÝZA POŽADAVKŮ NA SIMULÁTOR	12
2.1 SIMULACE TECHNOLOGICKÉHO PROCESU	12
2.2 TABULKY SIGNÁLŮ	14
2.3 ŘÍDICÍ ELEKTRONIKA	16
2.4 REKAPITULACE POŽADAVKŮ NA SIMULÁTOR	16
3. VIRTUÁLNÍ VERZE SIMULÁTORU	18
3.1 REALIZACE VIRTUÁLNÍHO SIMULÁTORU	18
3.2 VIRTUÁLNÍ MODEL SIMULÁTORU	19
4. NÁVRH A REALIZACE HW SIMULÁTORU	21
4.1 VÝBĚR KOMPONENT	21
4.2 ZAPOJENÍ DIGITÁLNÍCH VSTUPŮ	28
4.3 ZAPOJENÍ DIGITÁLNÍCH VÝSTUPŮ	30
4.4 ZAPOJENÍ ANALOGOVÝCH VSTUPŮ	32
4.5 ZAPOJENÍ ANALOGOVÝCH VÝSTUPŮ.....	32
5. REALIZACE MECHANICKÉ KONSTRUKCE SIMULÁTORU	34
5.1 REÁLNÁ VERZE SIMULÁTORU	34
6. IMPLEMENTACE ŘÍDICÍHO SOFTWARE SIMULÁTORU.....	37
6.1 TESTOVACÍ KÓD	38
6.2 SIMULACE FYZIKY	39
6.3 ČTENÍ A ZÁPIS SIGNÁLŮ.....	44
6.4 KNIHOVNY	45
7. KONFIGURAČNÍ ROZHRANÍ SIMULÁTORU	47
7.1 REALIZACE KONFIGURAČNÍ APLIKACE	47
7.2 DEMONSTRACE APLIKACE	48
7.3 PARAMETRY SIMULÁTORU	49
8. ZÁVĚR.....	51

SEZNAM OBRÁZKŮ

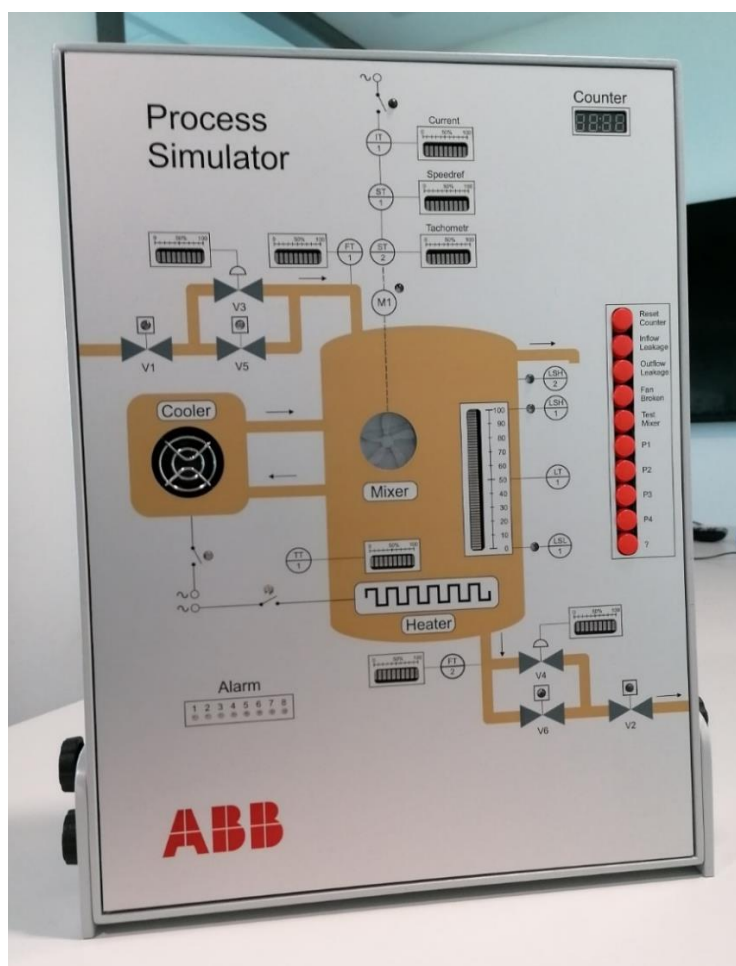
Obr. 1-1 – EUOPC OSLO Demo Simulátor	10
Obr. 2-1 – Blokový diagram technologického procesu	12
Obr. 2-2 – Schéma napouštění [1]	13
Obr. 2-3 – Schéma vypouštění [1]	13
Obr. 3-1 - Ukázka simulace – proces ohřívání a paralelní mixování	19
Obr. 3-2 - Virtuální model simulátoru bez montážní desky	20
Obr. 4-1 – Blokové schéma zapojení simulátoru	21
Obr. 4-2 – Arduino Mega 2560 [8]	22
Obr. 4-3 – Převodník úrovně napětí [9]	23
Obr. 4-4 - Reléový modul [10]	23
Obr. 4-5 - Převodník proudové smyčky na napětí [11]	23
Obr. 4-6 – Převodník PWM na 0-10 V [12]	24
Obr. 4-7 – PWM expander [13]	24
Obr. 4-8 - I/O modul [14]	25
Obr. 4-9 – Step down [15]	25
Obr. 4-10 – Malý bargraph modul [16]	26
Obr. 4-11 – Velký bargraph modul [17]	26
Obr. 4-12 - RGB LED modul [18]	26
Obr. 4-13 - RGB LED panel [19]	27
Obr. 4-14 – sedmisegmentový LED displej [20]	27
Obr. 4-15 – DC ventilátor [21]	27
Obr. 4-16 – DC motorek [22]	27
Obr. 4-17 – Řešení digitálního výstupu simulátoru [23]	28
Obr. 4-18 – Zapojení převodníků napětí	29
Obr. 4-19 – Řešení digitálního výstupu simulátoru [10]	30
Obr. 4-20 – Zapojení relé modulů	31
Obr. 4-21 – Zapojení převodníku proudové smyčky	32
Obr. 4-22 – Zapojení regulátoru typu L78M [24]	33
Obr. 4-23 – Zapojení převodníku PWM signálu	33
Obr. 5-1 – Konstrukce rámečku	34
Obr. 5-2 – Reálné zapojení přední strany simulátoru	35
Obr. 5-3 – Vnější panely	36
Obr. 6-1 – Blokové schéma řídicího SW	37
Obr. 6-2 - Simulace teploty v Matlabu	43
Obr. 6-3 Master-vysílač adresuje slave zařízení-přijímač [28]	44
Obr. 6-4 Struktura knihoven	46
Obr. 7-1 - Výpis ze sériového monitoru simulátoru	48
Obr. 7-2 - Realizace emulátoru Arduina v programu Proteus	48
Obr. 7-3 - Konfigurační aplikace	49

SEZNAM TABULEK

Tabulka 2-1 – vstupní signály simulátoru [4]	15
Tabulka 2-2 – výstupní signály simulátoru [4]	15

1. ÚVOD

EUOPC Oslo Demo simulátor je zařízení určené pro školící účely švédsko-švýcarské firmy ABB, jejíž hlavní zaměření je v oblastech elektrotechniky, pohonů, robotiky a průmyslové automatizace. Simulátor má sloužit převážně k testování funkčnosti řídicí aplikace, která je součástí školení T315 pro ABB systém 800xA. Funkcí aplikace je řídit technologický proces, který je také součástí školení. Tato aplikace, kterou v rámci kurzu T315 školené osoby naprogramují, je běžně testována na soft kontroléru. Je žádoucí, aby si účastníci kurzu uvědomili, že to, co funguje na soft kontroléru, nemusí nutně fungovat v reálném světě. K finálnímu ověření správného naprogramování aplikace slouží právě simulátor, který simuluje chování reálného systému v praxi.



Obr. 1-1 – EUOPC OSLO Demo Simulátor

Simulátor bude realizovat zadaný technologický proces, tedy bude simulovat chování reálné nádrže s kapalinou, která prochází jednotlivými fázemi procesu (napouštění, regulace teploty, mixování atd). Je požadováno, aby zpracovával vstupní analogové a digitální signály z I/O (input/output) karet ABB kontroléru a na základě těchto signálů

simuloval signály výstupní. Na předním panelu simulátoru bude použita grafika znázorňující simulační soustavu, přičemž přední část simulátoru bude současně sloužit i jako HMI (human-machine interface) panel. V reálném čase bude simulátor zobrazovat stavy akčních členů (mixér, chladič, topné těleso) a fyzikálních veličin (teplota, rychlost proudění, proud mixéru atd.), pomocí vhodných indikačních prvků. Současně bude možné simulovat poruchové a jiné signály pomocí tlačítek na HMI panelu.

Bude navržena vhodná elektronika pro provádění simulačních výpočtů a pro samotné řízení simulátoru. Elektronika bude příhodně upevněna v mechanické konstrukci simulátoru, která bude tvořena hliníkovým rámečkem s předním a zadním panelem. Konstrukce bude stát na stabilních podpěrách, které umožní pohodlnou práci se zařízením. Simulátor bude snadno demontovatelný, což je výhodné pro provedení případné opravy nebo přehrání SW (software).

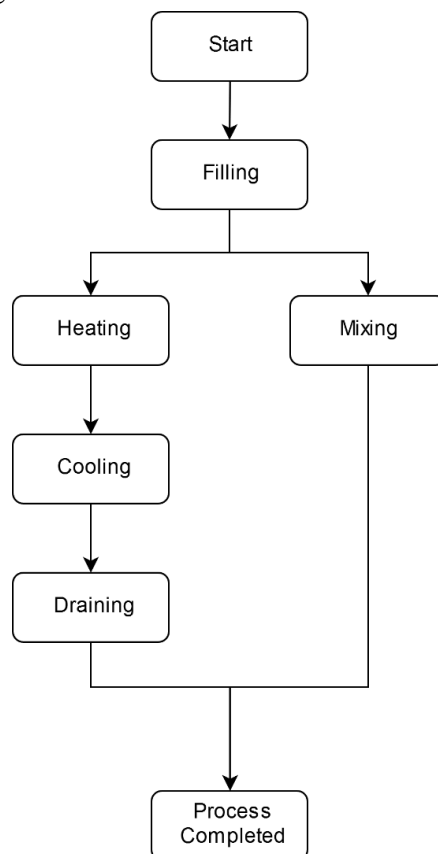
Vzhledem k tomu, že simulátor bude umístěn v ABB univerzitě, kde bude používán řadou lidí, bude další z požadavků reprezentativní vzhled celého zařízení. Napájení bude řešeno vhodným externím zdrojem, který se bude zapojovat ze zadní části simulátoru. V zadním panelu budou také výřezy pro konektory, kterými se simulátor bude připojovat k PLC (programmable logic controller) systému pomocí rozvodové svorkovnice. Tímto způsobem bude zajištěna zpětná kompatibilita s předchozími verzemi simulátorů.

2. ANALÝZA POŽADAVKŮ NA SIMULÁTOR

Kapitola pojednává o hlavních požadavcích kladených na simulátor. Konkrétně je zde rozebrána simulace technologického procesu a jednotlivých fyzikálních jevů, které se v simulaci realizují. Dále je popsána analýza požadavků na elektroniku simulátoru na základě hlavních požadavků firmy a tabulky signálů.

2.1 Simulace technologického procesu

Technologický proces, který má být simulován lze rozdělit na 5 hlavních podprocesů. Patří mezi ně napouštění, ohřev na požadovanou teplotu, ochlazení na požadovanou teplotu, vypouštění a mixování. Po dokončení procesu se simulátor vrátí do původního stavu a čeká na spouštěcí signál.

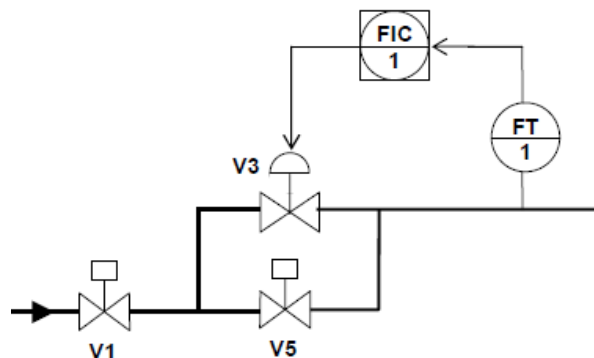


Obr. 2-1 – Blokový diagram technologického procesu

Simulace změny objemu kapaliny

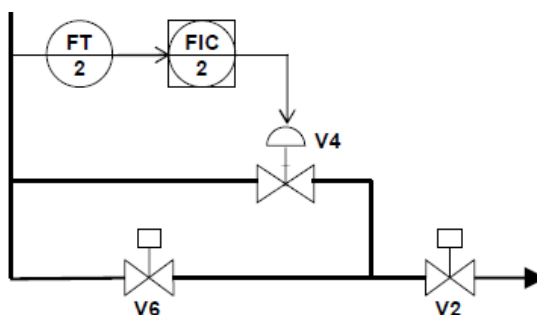
Napouštění, jehož schéma je uvedeno na Obr. 2-2, má být realizováno vstupním potrubím, regulační smyčkou a třemi ventily (V1, V3 a V5). Ventily V1 a V5 jsou digitální ventily, z toho ventil V5 funguje v omezeném režimu. I když je ventil V5 otevřený, omezuje celkový průtok kapaliny na 40 % a funguje jako obtokový – bypass ventil. Lineární ventil V3 představuje ovládací prvek, kterým lze průtok dále regulovat.

Regulační smyčka tvořena PID regulátorem (FIC1) a průtokoměrem FT1 (flow transmitter) však není předmětem simulace, ale řízení. Proto se jí v rámci této bakalářské práce není nutné zabývat. Bude však nutné vzít v potaz simulování průtoku, a tedy průtokoměr FT1.



Obr. 2-2 – Schéma napouštění [1]

Realizace vypouštění je podobná jako u napouštění, opět je použito potrubí, regulační smyčka a tři ventily. Ventily jsou totožné s těmi u napouštění, pouze jsou odlišně pojmenovány.



Obr. 2-3 – Schéma vypouštění [1]

Simulace změny teploty

Regulování teploty kapaliny na požadovanou hodnotu má být realizováno pomocí ohřívače, respektive chladiče, který má být řízen digitálním signálem. Požadavky na výkony těchto dvou zařízení nejsou nijak blíže specifikovány, nicméně pro jednoduchost bude uvažováno, že je výkon konstantní. Hodnota výkonu obou zařízení bude konfigurovatelná uživatelem prostřednictvím konfigurační aplikace.

Změna teploty kapaliny může nastat v různých případech. Může být způsobena cílenou regulací, nebo poruchou. Regulace teploty představuje sepnutí ohřívače nebo chladiče. Změna teploty se pak odvíjí od stavu obou zařízení. Porucha může být způsobena například přítokem nebo odtokem kapaliny při regulaci teploty. Přítokem kapaliny by došlo k mísení kapalin o různých teplotách a teplota kapaliny v nádrži by se mohla měnit. Odtokem kapaliny by se zase zmenšoval objem kapaliny v nádrži, jejíž teplota je regulována konstantně výkonným zařízením a docházelo by opět ke změně

teploty. Pro mísení kapalin dvou různých teplot bude k výpočtu využita kalorimetrická rovnice (rovnice 2.1). Poruchu také představuje únik tepla do okolí.

$$m_1 c_1 (t - t_1) = m_2 c_2 (t_2 - t) \quad [2](2.1)$$

Simulace mixování kapaliny

Mixování kapaliny je realizováno rotováním vrtule mixeru umístěného v nádrži. Mixér je v tomto případě pravděpodobně poháněn třífázovým motorem. Není přesně definováno, zda se jedná o asynchronní motor nebo synchronní, nicméně to v tomto případě nehraje až tak velkou roli, jelikož bude nutné pouze simulovat proud nutný k pohánění mixéru, tedy proud třífázového motoru. „Kritický ‘okamžik‘ představuje spouštění zatíženého motoru. Při rozběhu, kdy se otáčky zvyšují z nulové hodnoty na jmenovitou, odebírá motor mnohem větší proud než po ukončení rozběhu, tj. při jmenovitých otáčkách“[6]. Co se týče simulace proudu, bude nutné počítat s proudovou špičkou při rozběhu mixéru. Proud motoru se také bude měnit vzhledem k jeho zatížení. Bude-li tedy přibývat kapalina, poroste zátěžový moment, a tedy se bude muset zvyšovat proud, dle rovnice.

$$M = c \cdot \Phi \cdot I \quad [3](2.2)$$

Kde M je kroutící moment motoru, Φ je celkový magnetický tok a I je proud tekoucí kotvou. Na mixování budou také působit různé vnější vlivy, takže proud mixéru bude kolísat.

2.2 Tabulky signálů

V níže uvedených tabulkách jsou uvedeny vstupní a výstupní signály simulátoru. Pro digitální vstupy a výstupy musí simulátor pracovat s logikou signálu 0 V a 24 V. U analogových vstupů je požadavek na zpracování proudové smyčky, tedy 0-20 mA, nebo 4-20 mA. Analogový výstup simulátoru bude řešen napětovým signálem v rozsahu 0-10 V.

Tabulka 2-1 – vstupní signály simulátoru [4]

Simulator Inputs		
n	Digital Input	Analog Input
1	Heater on	Reference Mixer
2	Cooler on	Reference V3
3	Mixer on Power	Reference V4
4	V1 on	
5	V2 on	
6	V5 on	
7	V6 on	
8	Counter clock	
9	Counter up/down	
10	Counter reset	
11	Alarm 1	
12	Alarm 2	
13	Alarm 3	
14	Alarm 4	
15	Alarm 5	
16	Alarm 6	
17	Alarm 7	
18	Alarm 8	
19	Buzzer	
20	Heater on 40 %	

Tabulka 2-2 – výstupní signály simulátoru [4]

Simulator outputs		
n	Digital Output	Analog Output
1	Return Cooler	Temperature
2	Return Cooler (FT)	Mixer current
3	Return Heater	Total inflow
4	Return max level (LG H1)	Total outflow
5	Return min level (LT)	Level (Tank)
6	Return Mixer	Tachometer
7	Return Mixer power	
8	Return overflow (LG H2)	
9	Return V1 closed	
10	Return V1 open	
11	Return V2 closed	
12	Return V2 open	
13	Return V5 closed	
14	Return V5 open	
15	Return V6 closed	
16	Return V6 open	

2.3 Řídicí elektronika

Hlavním požadavkem na řídicí elektroniku je, aby byla schopna komunikovat s PLC systémem a spolehlivě simulovala chování simulované soustavy. Je tedy nutné, aby v reálném čase zpracovávala řídicí signály z PLC a na základě těch prováděla výpočty, jak by se chovala soustava. Fyzické spojení simulátoru s PLC bude realizováno dvěma 25 pinovými datovými kabely. Z I/O karet PLC systému budou vyvedeny jednotlivé signály na rozvodovou svorkovnici, ke které se připojí datové kabely skrze CANON 25 konektory. Tato konfigurace zapojení zajistí zpětnou kompatibilitu s předchozími verzemi simulátorů.

Samotný simulátor by měl být lehký, a ne příliš rozměrný, jelikož se bude přesouvat z místa na místo. Z tohoto důvodu je kladen důraz i na velikost řídicí elektroniky a její umístění v mechanické konstrukci. Tímto se nabízí řešení řízení pomocí mikrokontroléru, který výše zmíněná kritéria splňuje a jeho dalším plusovým parametrem je i cena, která je dalším z důležitých požadavků.

Je však nutné vzít v potaz, že úroveň signálu PLC systému a mikrokontroléru je různá. Bude třeba vyřešit, jak převést průmyslový standard (0-24 V, 4-20 mA atd.) na signálovou úroveň mikrokontroléru (0-5 V) a to pro všech 45 signálů, které jsou uvedeny v kapitole *Tabulky signálů*.

Požadavek je taky na jednoduché napájení celého systému, nejlépe ze síťové zásuvky. Tomuto požadavku pravděpodobně nejlépe vyhoví externí síťový adaptér, který se používá například pro napájení notebooků. Bude však nutné spočítat jaký výkon je potřeba pro napájení simulátoru.

Dále bude nezbytné, aby simulátor zobrazoval v reálném čase požadované veličiny na předním panelu, skrze RGB (red, green, blue) LED (Light-Emitting Diode) diody, LED bargrafy, sedmisegmentové displeje apod. Řídicí elektronika tedy bude muset být uzpůsobena pro ovládání těchto periférií. Bude tedy muset disponovat komunikací přes rozhraní I2C (Inter-Integrated Circuit) nebo SPI (Serial Peripheral Interface), přes které se obvykle řeší komunikace s těmito perifériemi. Pro řízení RGB diod bude zase nutná pulzně šířková modulace (PWM). Na předním panelu budou rovněž umístěny tlačítka pro uživatele.

2.4 Rekapitulace požadavků na simulátor

Zařízení musí simulovat technologický proces, který se skládá z pěti hlavních částí viz Obr. 2-1. Po ukončení technologického procesu je nutné, aby se simulátor vrátil do původního stavu a čekal na spouštěcí signál. Simulátor musí zpracovávat v reálném čase vstupní signály z PLC, a na základě nich provádět výpočty výstupních signálů. Tyto výstupní signály mají být přivedeny na vstupní karty PLC systému, přičemž propojení simulátoru a PLC bude provedeno dvěma CANON konektory. Je tedy nezbytné, aby byl simulátor kompatibilní s rozsahy signálů PLC karet. Pro digitální vstupní i výstupní

signály jde o logiku 0 V a 24 V. Pro analogové vstupní signály jde o napěťový rozsah 0-10 V a pro analogové výstupní signály je to proudová smyčka 0-20 mA. Vstupní i výstupní signály jsou definovány v tabulkách Tabulka 2-1 a Tabulka 2-2.

Řídicí elektronika simulátoru musí být spolehlivá a neměla by být příliš drahá. Je nutné, aby kromě řízení simulátoru byla schopna obsloužit i indikační prvky umístěné na předním panelu, které budou zobrazovat požadované veličiny simulátoru. Přední panel má současně sloužit i jako HMI panel a je požadavek, aby přes něj bylo možné simulátor ovládat. Dále je požadavek, aby napájení simulátoru a řídicí elektroniky bylo jednoduché, například pomocí externího síťového zdroje.

Konstrukce simulátoru musí být lehká a zařízení nesmí být příliš rozměrné. Je totiž pravděpodobné, že bude simulátor přenášen z místa na místo. Důraz je také kladen na design předního panelu simulátoru.

3. VIRTUÁLNÍ VERZE SIMULÁTORU

Kapitola popisuje tvorbu virtuálního simulátoru v systému ABB 800xA. Ten je použit k otestování správné simulace technologického procesu bez nutnosti připojení k reálnému PLC systému. Cílem je komplexnější pochopení řešeného problému, které se pozitivně projeví v návrhu a výběru elektroniky. Výhodou je i funkční algoritmus realizující fyziku simulátoru, který bude dále jen implementován v jiném programátorském prostředí.

V kapitole je rovněž popsán návrh virtuálního modelu simulátoru. Ten je použit pro určení rozměrů montážní desky a rámečku simulátoru. Díky virtuálního modelu je známá i přibližná pozice modulů na montážní desce, a tedy i pravděpodobné vedení kabeláže.

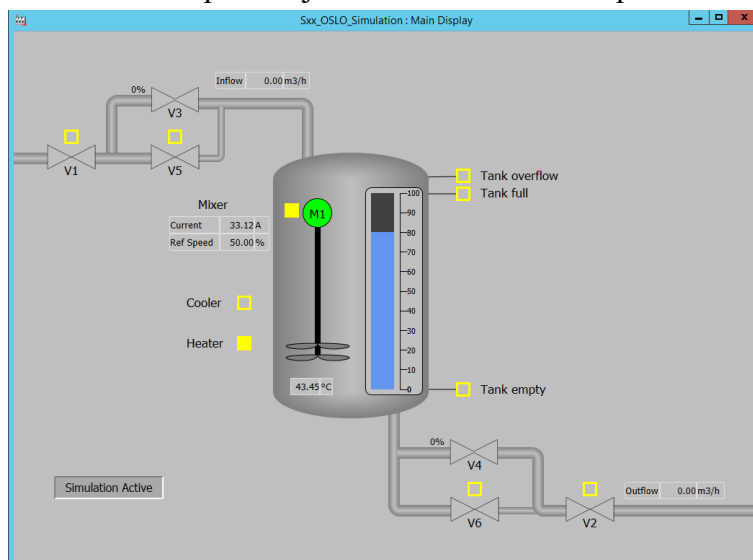
3.1 Realizace virtuálního simulátoru

Před samotným návrhem zařízení, byla vytvořena jeho virtuální verze, která obsahovala jednoduché grafické rozhraní simulátoru a základní algoritmus simulace technologického procesu. Virtuální simulátor sloužil k ověření fungování fyziky simulátoru, na základě aktivací vstupních signálů. Tímto bylo docíleno ověření fungování algoritmu, který realizoval simulaci technologického procesu (další podrobnosti k algoritmu jsou uvedeny v kapitole 6.2). Tento algoritmus už tedy mohl být připraven na implementaci v reálném zařízení. Z hlediska programování by už zbývala pouze část kódu, která sloužila k realizaci komunikace řídicí elektroniky. Virtuální kopie simulátoru byla realizována pomocí systému ABB 800xA.

Systém 800xA je komplexní řídicí systém, který integruje funkcionalitu inženýringu, operativního řízení, řízení I/O rozhraní, bezpečnosti, dat či receptur. Síla systému spočívá ve spojení všech uvedených funkcionalit, systém tedy umožňuje vytvořit kvalitní kolaborativní prostředí pro projektové řízení. Toto kolaborativní řízení je výhodné z pohledu dostupnosti informací a dat napříč systémem. Vezmeme-li například ventil jako objekt, který je předmětem řízení v tomto projektu simulace. Systém 800xA vytváří tomuto objektu kontejner dat – aspektů, které blíže specifikují daný ventil (například grafika ventilu, alarmový a událostní seznam, výkresová dokumentace, popis parametrů atd.) [5]. V tomto systému jsou tedy řešeny projekty pro řízení všech možných procesů. Rovněž je v něm realizována řídicí aplikace pro Oslo Demo simulátor. Tato aplikace měla řídit požadovaný technologický proces a byla naprogramována školenou osobou. Řídicí aplikace není předmětem této bakalářské práce, ale souvisí s navrhovaným simulátorem. Bude použita pro otestování funkčnosti virtuálního a poté i reálného zařízení.

Virtuální simulátor byl realizován jako externí knihovna 800xA, která mohla být jednoduše vložena do řídicí aplikace. Virtuální simulátor nebyl fyzicky spojen s PLC systémem, ale bylo zde využito namapování signálů na soft PLC systému 800xA. Tímto bylo možné aktivovat vstupní signály simulátoru z virtuálních I/O karet soft PLC.

Jednoduchá grafika, která je zobrazena na Obr. 3-1, je tvořena nádrží se sloupcovým indikátorem aktuálního množství média a informací o jeho teplotě. Součástí nádrže je také mixér s binárním ukazatelem jeho stavu a ukazateli proudu a rychlosti otáček. Nádrž po stranách také obsahuje binární ukazatele zobrazující stavy chladiče, topného tělesa a tří plovákových spínačů. Součástí grafiky je také vstupní a výstupní potrubí s ventily a indikací jejich stavů. Pro každé potrubí je zobrazována hodnota průtoku.



Obr. 3-1 - Ukázka simulace – proces ohřívání a paralelní mixování

Tvorba virtuálního simulátoru byla užitečná z hlediska ověření funkčnosti algoritmu pro fungování fyziky simulátoru. Rovněž sloužila k ujasnění si signálů nutných pro tvorbu simulátoru. Pro správné fungování simulátoru je nutné, aby například signál Return V1 open, uvedený v tabulce výstupních signálů simulátorů figuroval jako digitální výstup pro PLC. Zároveň je však nutné tento signál použít pro indikaci stavu ventilu V1 v grafice, respektive na předním panelu. Tento vstupní signál jsou tedy ve skutečnosti dva signály. Tohle zjištění je kritické pro další návrh a výběr elektroniky simulátoru

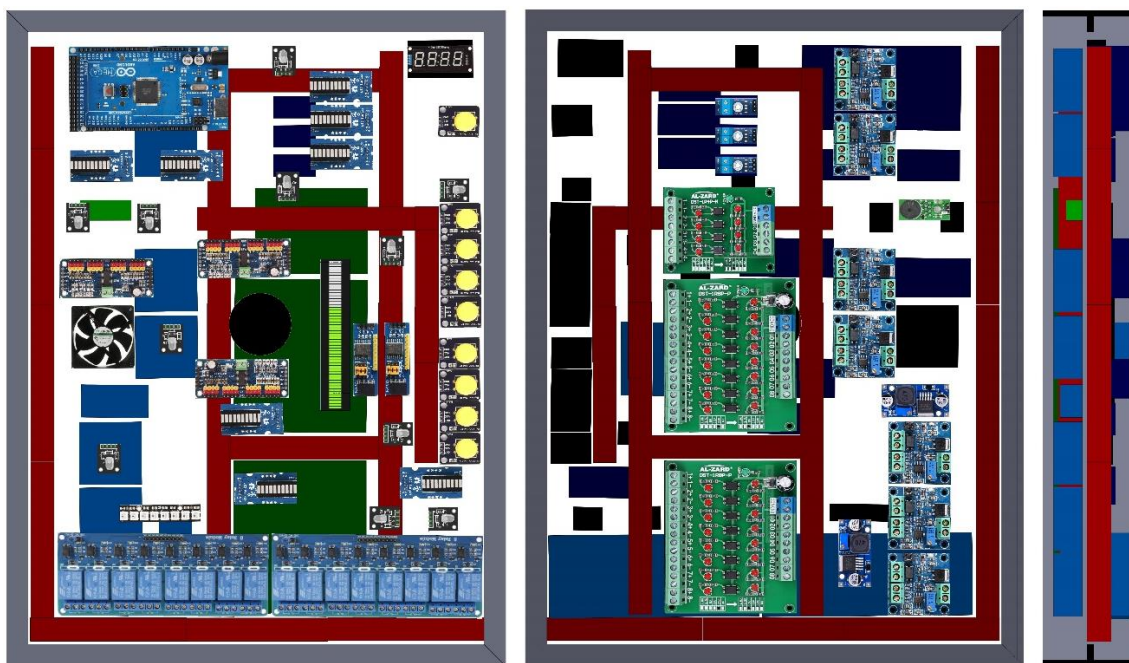
3.2 Virtuální model simulátoru

Podkapitola popisuje návrh virtuálního modelu simulátoru na základě informací uvedených v kapitole *Návrh a realizace HW simulátoru*. Tato podkapitola tedy chronologicky nenavazuje na předchozí text, nicméně svým obsahem zapadá pod téma virtuální verze simulátoru. Z chronologického hlediska postupu návrhu simulátoru lze tuto podkapitolu zařadit mezi kapitoly *Návrh a realizace HW simulátoru* a *Realizace mechanické konstrukce simulátoru*.

Před návrhem samotné mechanické konstrukce, byl ve vývojovém prostředí SolidWorks vytvořen virtuální model fyzického simulátoru (Obr. 3-2). V tuto chvíli byly známy jednotlivé rozměry a počet modulů řídicí elektroniky, a tudíž je bylo možné namodelovat a virtuálně je rozmísťovat, za účelem co nejlepšího využití plochy. Zásadní

bylo rozmístění indikačních prvků, které jsou vidět a jejich umístění je víceméně fixní. S ostatními moduly bylo možné pohybovat v podstatě libovolně, v úvahu se brala pouze jejich výška (některé moduly je výhodnější umístit na přední stranu z důvodu většího prostoru). Mimo efektivního využití plochy se uvažovala i realizace vedení kabeláže a mechanické uchycení modulů. Dává smysl, aby moduly se stejnou funkcí byly umístěny blízko sebe, jelikož k nim pravděpodobně povedou vodiče ze stejného konektoru. Tímto je docíleno toho, že vodiče mezi modulem a konektorem mají podobnou délku a vliv úbytku napětí ve vodiči lze zanedbat. Mimo přesných rozměrů modulů byly známa i poloha jejich montážních otvorů. Takto bylo možné kontrolovat, že se montážní otvory nepřekrývají, a že uchycení modulu A nekoliduje s modulem B na druhé straně montážní desky.

Návrhem virtuálního modelu simulátoru bylo docíleno zjištění přesných rozměrů potřebných pro montážní desku. Rovněž byla zjištěna požadovaná hloubka simulátoru, a tedy bylo možné vytvořit model a výkres rámečku do kterého se elektronika simulátoru usadí. V neposlední řadě se zjistilo pravděpodobné rozmístění modulů a vedení kabeláže, což velice usnadní samotné sestavení simulátoru.

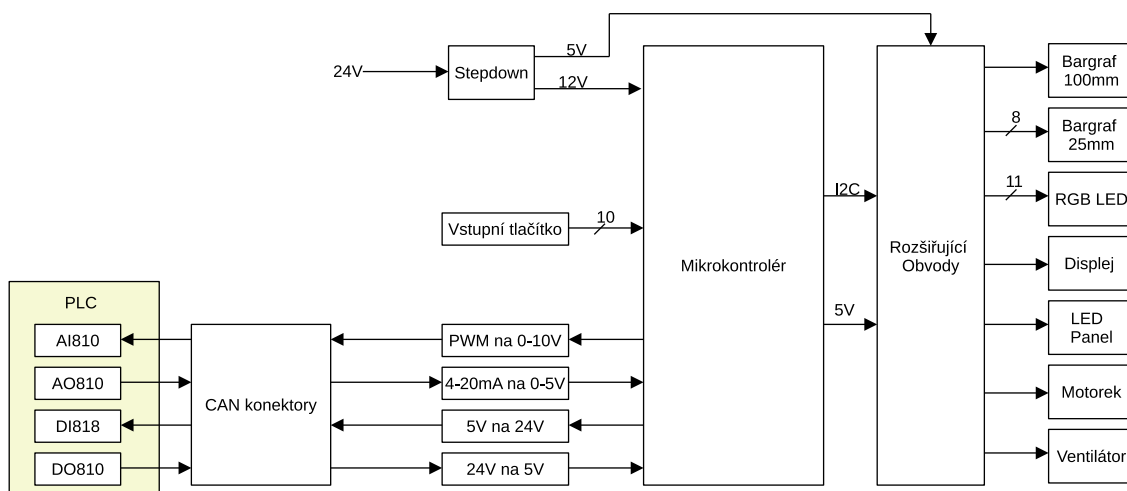


Obr. 3-2 - Virtuální model simulátoru bez montážní desky

4. NÁVRH A REALIZACE HW SIMULÁTORU

Tématem kapitoly je výběr elektroniky, základní zdůvodnění výběru a stručný popis jejího fungování. Kapitola se dále věnuje detailnímu řešení vstupů a výstupů simulátoru. Vysvětluje fungování elektroniky realizující propojení mikrokontroléru s PLC systémem a zobrazuje její reálné zapojení. Přehledné zapojení elektroniky je zobrazeno v blokovém schématu na Obr. 4-1

V levé části obrázku jsou zelenou barvou vyznačeny I/O karty modulárního PLC, ke kterým je simulátor připojen. Propojení PLC a simulátoru je realizováno dvěma CANON konektory, které jsou nakresleny pro zjednodušení jedním blokem. Jelikož jsou rozsahy I/O karet a mikrokontroléru odlišné, je nutné signály převést na požadovanou úroveň. K tomu slouží čtyři typy převodníků, které jsou zapojeny mezi CANON konektory a piny mikrokontroléru. Vzhledem k velkému počtu obsluhovaných periférií jsou k mikrokontroléru připojeny další rozšiřující obvody, které jsou pro zjednodušení nakresleny v jednom bloku. Tyto obvody komunikují skrze I2C protokol a rozšiřují I/O piny mikrokontroléru. Tímto způsobem jsou ovládány indikační moduly v pravé části (bargrafy, RGB LED diody atd). Napájení simulátoru je ze síťového zdroje o velikosti napětí 24 V, přičemž snížení napětí pro mikrokontrolér a rozšiřující modely je řešeno stepdown moduly.



Obr. 4-1 – Blokové schéma zapojení simulátoru

4.1 Výběr komponent

U výběru komponent je kladen důraz na bezpečné spojení simulátoru s PLC systémem tak, aby při poruše nebyl poškozen mikrokontrolér. Piny mikrokontroléru nemusí zvládnout to, co by zvládl robustní PLC systém. Proto je nutné zajistit galvanické oddělení pro citlivé piny mikrokontroléru, například pomocí optočlenu. Dalším důležitým požadavkem na výběr elektroniky je samozřejmě její spolehlivost a cena.

Mikrokontrolér

Pro řízení byl zvolen osmibitový mikroprocesor ATmega2560, který je součástí vývojového kitu Arduino Mega 2560. Mikroprocesor disponuje 256 kB programovatelné paměti FLASH, 8 kB statické paměti SRAM a 4 kB paměti EEPROM. Pracuje na frekvenci 16 MHz a programuje se přes USB (universal serial bus) konektor. Dále obsahuje 54 digitálních vstupně výstupních pinů, přičemž 15 z nich je konfigurovatelných na PWM výstup, mimo to je k dispozici i 16 analogově vstupních pinů. Operační napětí mikrokontroléru je 3,3–5 V. Maximální výstupní proud je 20 mA na 3,3 V pin a 50 mA na 5 V pin. K dispozici jsou také komunikační piny skrze rozhraní I2C, SPI nebo UART [8].

Napájení Arduina je možné realizovat třemi možnými způsoby. První možností je přes USB port, který se používá i k programování a jeho napájecí napětí je v rozmezí 4,75–5,25 V. Maximální možný proud se pak pohybuje kolem 500 mA. Druhým způsobem je připojení přes svorky VIN a GND, kdy se napájecí napětí pohybuje v rozmezí 5,8–14,8 V a maximální proud se zdvojnásobuje na 1000 mA. Bohužel v tomto provedení není realizována ochrana proti přepólování. Poslední variantou, která je i tou, která byla zvolena, je napájení přes souosý konektor, kdy se napětí pohybuje od 6 do 15 V a maximální proud nabývá hodnoty 1000 mA [7]. Při této možnosti je dosaženo nejvyššího proudového výstupu, ochrany proti přepólování, a hlavně je zajištěno pevné spojení díky souosému konektoru. Tím pádem nehrozí, že by se napájení odpojilo, například pokud by simulátor spadl.

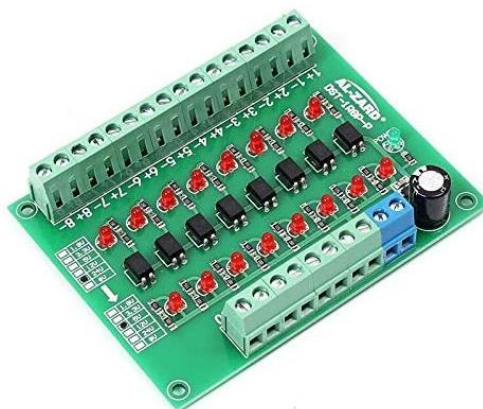
Výhodou zvoleného modulu je snadné nahrávání programu přímo z PC (personal computer) skrze USB rozhraní, množství a konfigurovatelnost I/O pinů, bohatá sada přídatných periférií, jednoduchá výměna a možnost upevnění modulu.



Obr. 4-2 – Arduino Mega 2560 [8]

Digitální vstupy

Pro převod digitálního signálu z PLC na požadovanou úroveň napětí pro Arduino byly použity převodníky s optočlenem. Tyto moduly se vyrábí ve čtyřkanálové a osmikanálové variantě. Modul je napájen napětím o velikosti 5 V a na základě logické úrovně 24V signálu na vstupu je generována 5V logická úroveň na výstupu.



Obr. 4-3 – Převodník úrovně napětí [9]

Digitální výstupy

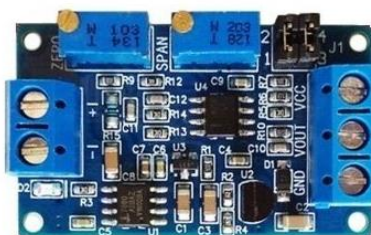
Pro spínání 24V signálu menším 5V signálem se ideálně hodí relé. Proto zde byl použit reléový modul s galvanickým oddělením, který zajistí ochranu pro mikrokontrolér. Relé se spínají uzemněním příslušného pinu a obsahují i světelnou indikaci sepnutí.



Obr. 4-4 - Reléový modul [10]

Analogové vstupy

Výstup analogových karet PLC je konfigurovatelný, ovšem pouze v rámci proudové smyčky. Tento signál Arduino zpracovávat neumí, proto bylo nutné použít převodník proudové smyčky na napěťový výstupní signál. Výhoda tohoto modulu je v tom, že je také konfigurovatelný. Pomocí potenciometrů lze nastavit, zda se chce zpracovávat signál 0-20 mA nebo 4-20 mA. Mimo proudového výstupu lze nastavit i výstupní napětí.

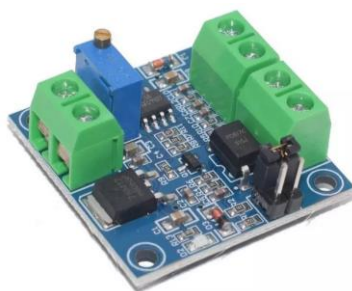


Obr. 4-5 - Převodník proudové smyčky na napětí [11]

Analogové výstupy

Nevýhodou vývojového kitu Arduino Mega 2560 je absence pinů realizující pravý analogový výstup. Výstup pro analogovou veličinu je zde řešen pomocí PWM signálu, který bohužel není kompatibilní s PLC. Proto byl použit převodník z PWM signálu na analogový signál napětí 0-10 V.

Napájecí napětí modulu je v rozmezí 12-30 V a výstupní napětí je nastavitelné pomocí potenciometru. Modul pracuje v rozmezí frekvencí 1-3 kHz. Velkou výhodou tohoto modulu je jeho převodní poměr, kdy PWM signál 0-100 % přímo odpovídá analogové hodnotě 0-10 V.

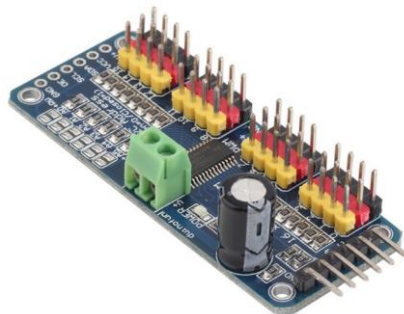


Obr. 4-6 – Převodník PWM na 0-10 V [12]

PWM expander

I přesto, že Arduino obsahuje digitální piny, které lze konfigurovat na generování PWM signálu, jejich počet nedosahuje požadavkům. Je nutné, aby Arduino bylo schopno obsloužit 12 RGB LED modulů, 9 LED bargrafů a sedmisegmentový LED displej. Samozřejmě je nutné vzít v potaz i výstupní analogové signály. Je třeba proto Arduino rozšířit o další PWM piny.

K tomuto slouží PWM modul, který se primárně používá pro připojování servo motorů. Z tohoto důvodu obsahuje i piny pro externí napájení modulů, což je velice vhodné, protože připojením RGB diod a dalších indikačních modulů nebude zatížen proudový výstup mikrokontroléru. Připojením jednoho modulu přes I2C rozhraní je Arduino rozšířeno o 16 dalších PWM pinů.

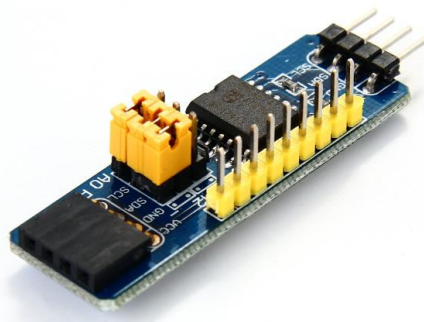


Obr. 4-7 – PWM expander [13]

I/O expander modul

Jelikož bude v řídicí aplikaci nutné obsloužit poměrně velký počet digitálních I/O signálů, je nezbytné rozšířit Arduino o I/O expander moduly. Tento modul se ke kontroléru připojuje přes I2C sběrnici a umožňuje na ni adresovat až 8 zařízení.

Deska obsahuje 8 konfigurovatelných I/O pinů, které lze softwarově nastavit na vstupní nebo výstupní. Zapojením modulů za sebe je možné mikrokontrolér rozšířit až o 64 dalších I/O pinů. Připojení je provedeno skrze 2 napájecí piny +5 V, GND a 2 datové piny SCLK a SDI.

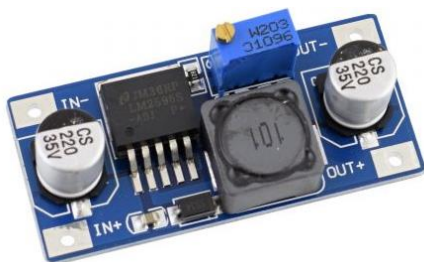


Obr. 4-8 - I/O modul [14]

Step down modul

Pro správné fungování digitálních výstupů simulátoru a jejich propojení s PLC je nutný zdroj napětí +24 V. Toto napětí ovšem není vhodné jako napájení pro řídicí elektroniku. Mikrokontrolér a periferie požadují nižší hodnotu napájecího napětí a z tohoto důvodu je nezbytné použít step down modul.

Ten funguje na principu řídicího čipu LM2596 a zajišťuje transformaci vyššího napětí na nižší, s účinností 92 %. Dle výrobce je definován maximální výstupní proud modulu 3 A, což je pro simulátor vyhovující. Výhodou je nastavení výstupního napětí pomocí potenciometru. Tímto způsobem je možné převést vstupní napětí v rozsahu 5-35 V na výstupní napětí 3-33 V.



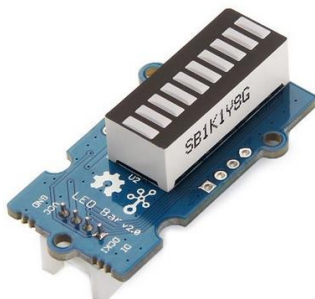
Obr. 4-9 – Step down [15]

Indikační moduly

Pro zobrazování analogových veličin se ideálně hodí LED bargrafy. Ty budou přehledně zobrazovat hodnotu průtoku, stav otevření lineárního ventilu, teplotu nebo objem kapaliny v nádrži. Pro realizaci simulátoru byly zvoleny dva typy bargraf modulů. Menší modul obsahuje bargraf s deseti samostatně ovládatelnými LED diodami. Deska modulu

obsahuje i řídicí čip, který usnadňuje řízení bargrafu. Modul je napájen +5 V a řízen dvěma digitálními signály DCKI a DI.

Podobným způsobem je řešen i větší modul, který je použit pro zobrazení objemu média v nádrži. Oproti menšímu modulu obsahuje bargraf 50 samostatných LED diod.



Obr. 4-10 – Malý bargraph modul [16]



Obr. 4-11 – Velký bargraph modul [17]

K indikaci stavů digitálních signálů byla zvolena RGB LED dioda. Výhoda oproti obyčejné LED diody je hlavně to, že je možné názorněji zobrazovat stavy digitálních ventilů. Je požadavek, aby v zavřeném stavu ventil svítil červeně, zatímco v otevřeném zeleně. Současně je možné navolit jakoukoliv barvu nebo jas diody pomocí 3 PWM signálů. Tímto je docílena možnost konfigurace indikačních LED přímo na míru simulátoru.



Obr. 4-12 - RGB LED modul [18]

Dále je nezbytné zobrazovat alarmové situace při řízení technologického procesu. K tomu slouží LED panel, který obsahuje 8 RGB diod. Každá dioda svým rozsvícením reprezentuje aktivaci jednoho z alarmů. Díky technologii RGB je opět možná konfigurace barvy a svítivosti na míru. Modul obsahuje 2 napájecí piny +5 V, GND a 2 řídicí piny DCKI a DI. Mimo panelu lze alarmové situace indikovat i akustickým

bzučákem, který je rovněž součástí simulátoru.



Obr. 4-13 - RGB LED panel [19]

Technologický proces je v kapitole 2.1 definován blokovým diagramem. Při dokončení procesu je v řídicí aplikaci vygenerován signál pro inkrementování počtu vykonaných procesů. K zobrazení tohoto čísla slouží sedmissegmentový LED displej se společnou anodou. Modul obsahuje řídicí čip TM1637 a je řízen dvěma digitálními signály DCKI a DI. Napájení je 3,3-5 V.



Obr. 4-14 – sedmissegmentový LED displej [20]

Stav chladiče simulátoru reprezentuje malý ventilátor o rozměrech 40x40x10 mm. Napájení ventilátoru je 5 V a jmenovitý proud 94 mA.



Obr. 4-15 – DC ventilátor [21]

Mixér je realizován malým DC motorkem typu RF-300CA, na jehož hřídeli je vrtulka vytisknutá na 3D tiskárně. Pracovní napětí motorku je 1,5-6 V a proud naprázdno činí 22 mA. Motorek bude připojen na PWM výstup mikrokontroléru, přičemž jeho otáčky budou řízeny jednoduše pomocí změny napájecího napětí.

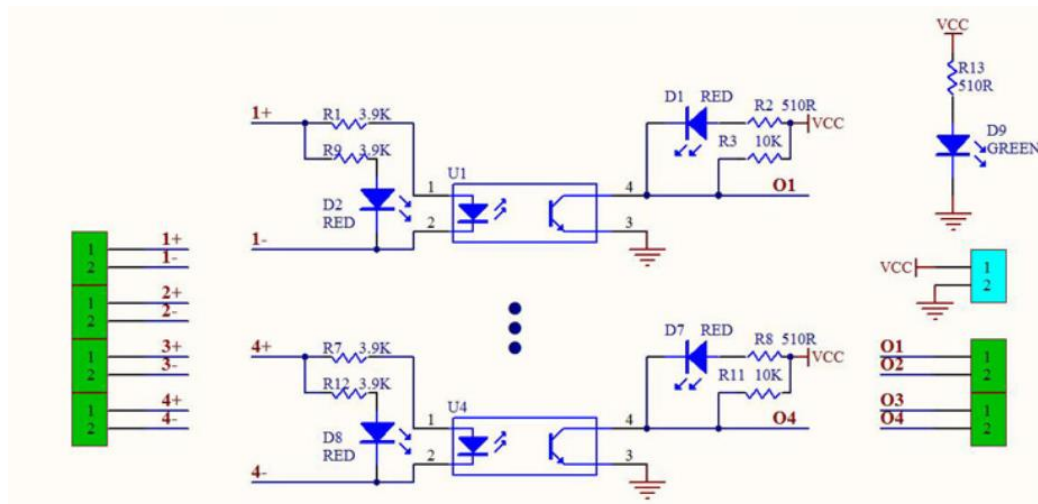


Obr. 4-16 – DC motorek [22]

Na předním panelu simulátoru se bude vyskytovat i 10 testovacích tlačítek. Ta budou sloužit k manuálnímu ovládní simulátoru.

4.2 Zapojení digitálních vstupů

Jak již bylo zmíněno v podkapitole výběru elektroniky, digitální vstup simulátoru je řešen převodníkem úrovně napětí. Ten zajišťuje bezpečné propojení PLC systému s mikrokontrolérem. Konkrétní řešení zpracování digitálního vstupu na elementární úrovni je znázorněno na Obr. 4-17.



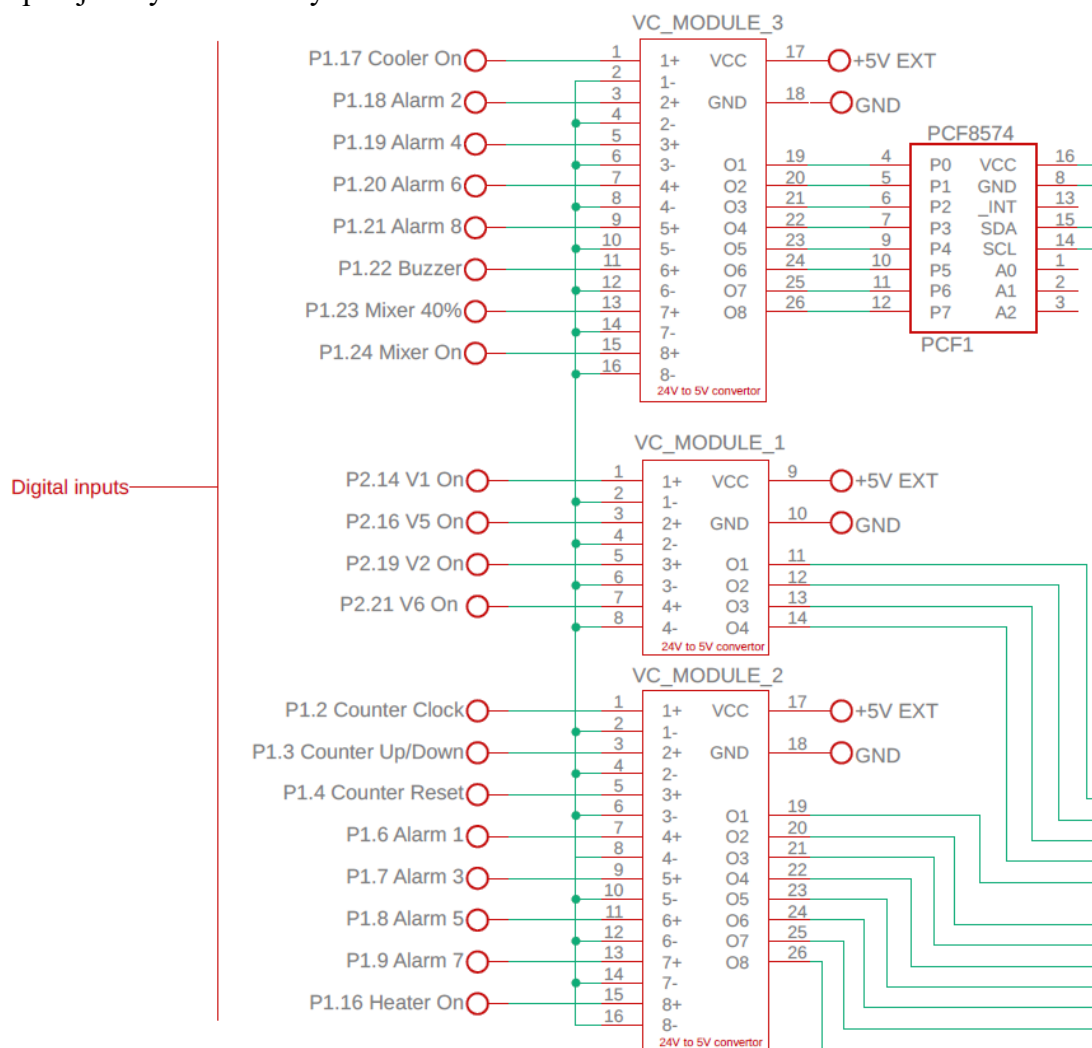
Obr. 4-17 – Řešení digitálního výstupu simulátoru [23]

Obvod se skládá z napájecí, vstupní a výstupní části. Požadovaný signál je přiveden vodičem opatřeným zakončovací dutinkou a připojen do svorkovnice modulu na příslušný pin. Spínání výstupního pinu O1 na základě vstupního 1+ zajišťuje optočlen. Obvody tedy nejsou galvanicky propojeny. Tím je zajištěna ochrana mikrokontroléru při možné poruše. Modul je napájen +5 V a toto napětí je přivedeno na svorku VCC. Indikace přítomnosti napětí je realizována zelenou LED diodou. Předřadný odpor R13 o hodnotě 510 ohmů zajišťuje takový proud, aby dioda nebyla zničena. V tomto případě jde o proud 9,8 mA.

Zemní svorky vstupních signálů (1-, 2-...) jsou spojeny a připojeny k zemi napájecího zdroje simulátoru. Vstupní signál je tedy aktivován pouze přivedením napětí na kladné svorky vstupních signálů. Ty jsou přímo propojeny k PLC systému skrze CANON konektor a rozvodovou svorkovnici. Vstupní svorky modulu jsou tedy fyzicky spojeny s výstupní DO kartou PLC systému. Přivedením napětí +24 V na kladnou svorku začne vstupním obvodem protékat proud. Napětí na odporu R1 a R9 bude rovno zhruba 22 V. Pokles 2 V oproti vstupnímu napětí je zapříčiněn úbytkem napětí na červené LED diodě indikující aktivaci vstupní svorky a samotné LED diodě optočlenu. Přesná hodnota úbytku napětí na LED diodách závisí na jejich technologickém provedení. Vstupní obvod je tvořen dvěma paralelními větvemi tvořenými odporem o hodnotě 3,9 k Ω a LED diodou. Pro zjednodušení lze předpokládat, že červená LED dioda má podobné parametry jako dioda optočlenu. V tomto případě se celkový proud rozdělí na dva stejně velké

proudy. Tento poloviční proud rozsvítí diodu optočlenu, který přes sepnutý fototranzistor uzemní výstupní obvod.

Je patrné, že při aktivaci vstupního pinu je výstupní pin v hodnotě LOW. Proud tekoucí výstupním obvodem rozsvítí LED diodu indikující sepnutí výstupu. Rezistor R3 s odporem 10 k Ω funguje jako pull-up rezistor a v tomto případě zabráňuje zkratování výstupního obvodu. Při nulovém napětí na svorce 1+ je naopak fototranzistor uzavřen a na výstupní svorce O1 je napájecí napětí +5 V tedy úroveň HIGH. Výstup převodníku napětí je tedy invertovaný.



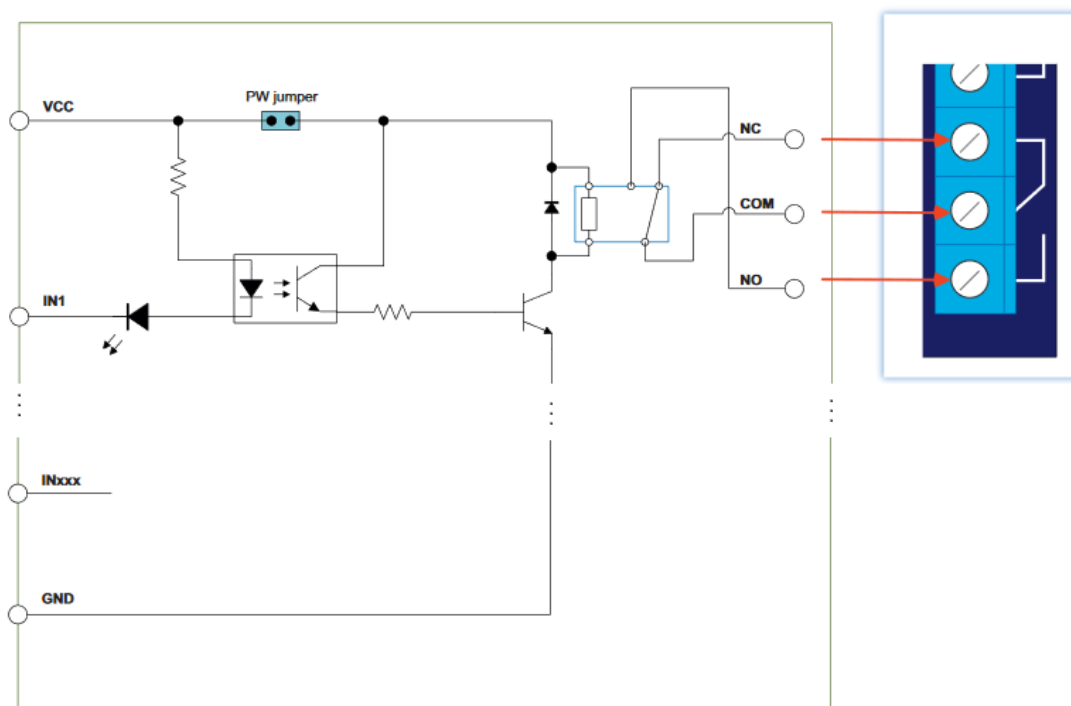
Obr. 4-18 – Zapojení převodníků napětí

Obr. 4-18 ukazuje realizaci zapojení všech 20 digitálních vstupů. Digitální vstupy jsou značeny jako číslo CANON konektoru, číslo jeho pinu a názvem signálu, který je tímto způsobem realizován (např. P1.17 Cooler On). Vstupy jsou přímo přivedeny na převodníky napětí. Pro zpracování výstupních signálů z převodníků je použit jeden I/O expander (PCF1), který je s Arduinem komunikuje pomocí I2C sběrnice. Zbylé signály jsou zpracovány analogovými vstupy mikrokontroléru. Zpracování digitálních signálů pomocí analogových vstupních pinů může znít jako plýtvání potenciálem

mikrokontroléru, nicméně z 16 vstupních analogových pinů jsou pro analogové vstupy využity pouze 3. Zbýlých 13 pinů by zůstalo nevyužitých, a tak byla možnost je zužít tímto způsobem.

4.3 Zapojení digitálních výstupů

Digitální výstupy simulátoru jsou řešeny pomocí relé modulů. Moduly opět využívají optočleny a zajišťují galvanické oddělení mikrokontroléru od PLC. Dle datasheetu zvládne relé spínat stejnosměrné napětí o velikosti 250 V a proud 10 A, což je pro potřeby simulátoru více než dostačující. Toto relé je napájeno napětím 5 V a spotřebovává okolo 0,36 W. Z toho vyplývá, že pro napájení cívky jednoho relé je potřebný proud zhruba 72 mA. Vzhledem k tomu, že digitálních výstupů je 16, je možné, že při teoretickém sepnutí všech digitálních výstupů najednou, by celkový napájecí proud pouze pro relé moduly představoval 1,125 A. Tento proud je příliš velký na to, aby mohl být dodáván z přímo z mikrokontroléru, a proto je nezbytné relé moduly připojit k jinému zdroji napájení. Není pravděpodobné, že by nastala situace, kdy budou všechna relé sepnuta, nicméně je třeba s tímto proudem počítat při návrhu a výběru zdroje.

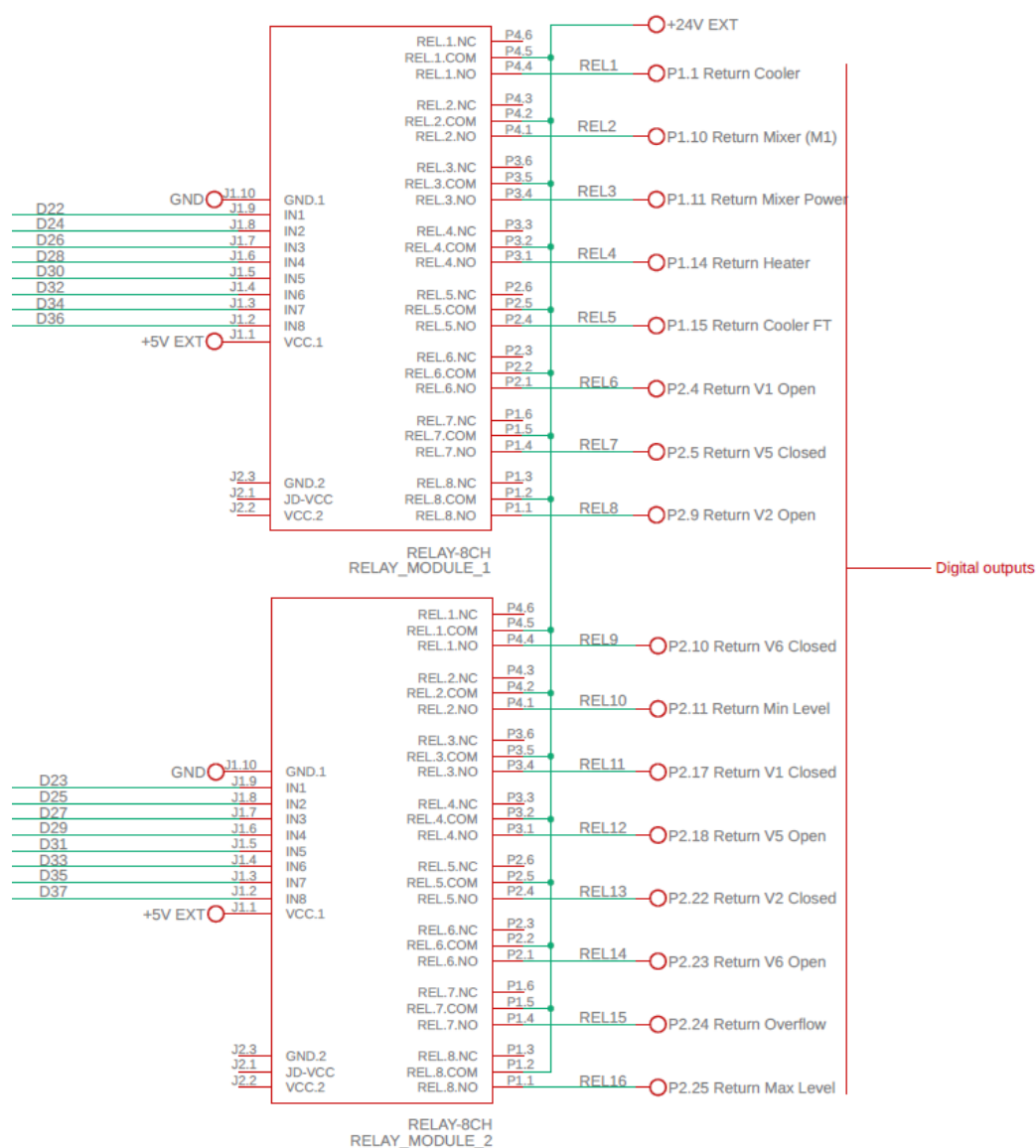


Obr. 4-19 – Řešení digitálního výstupu simulátoru [10]

Na Obr. 4-19 je zobrazeno detailní zapojení představující realizaci digitálního výstupu simulátoru. Na svorku VCC je připojeno napětí 5 V z externího zdroje. Na svorku relé COM je připojeno napětí 24 V, které bude při aktivaci vstupu přivedeno na svorku NO (normally open) a tedy i na výstup simulátoru. NO svorky z relé jsou přímo propojeny s CANON konektorem simulátoru. Relé se spíná uzemněním požadovaného vstupního

pinu. Je-li IN1 v logické nule, z VCC teče přes předřadný rezistor a LED diody proud. Tím se rozsvítí LED dioda optočlenu, která sepne fototranzistor. Proud tekoucí otevřeným fototranzistorem umožní sepnutí tranzistoru. Otevřeným tranzistorem se spojí obvod daného relé a jeho cívkou teče proud. Sepnutím relé dojde k přepnutí kontaktu a svorka NO se spojí se svorkou COM. Na výstupní pin CANON konektoru bude přivedeno napětí 24 V. Sepnutí relé je také indikováno rozsvícením červené LED diody.

Jak již bylo zmíněno výše, relé jsou napájena externím napětím o velikosti 5 V. Samotné spínání relé k logické nule realizuje mikrokontrolér pomocí digitálních pinů. Na Obr. 4-20 je vidět zapojení obou relé modulů a tedy všech 16 digitálních výstupů. Vodiče po levé straně označeny jako D22 až D37 představují připojení k digitálním pinům arduina D22-D37. Výstupy z relé modulů jsou značeny jako číslo CANON konektoru, číslo jeho pinu a názvem signálu, který je tímto způsobem realizován (např. P2.9 Return V2 Open).

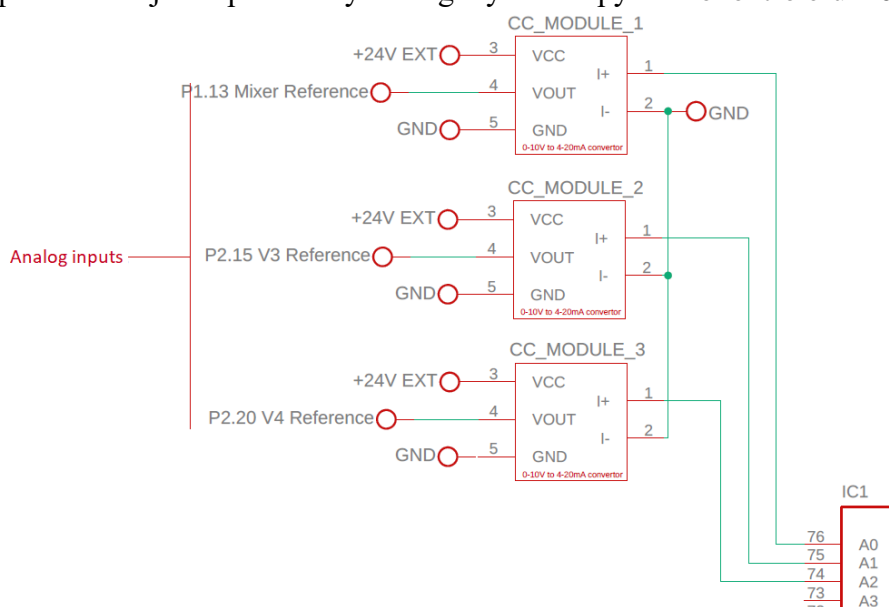


Obr. 4-20 – Zapojení relé modulů

4.4 Zapojení analogových vstupů

Propojení analog output (AO) karet PLC systému s mikrokontrolérem realizuje převodník proudové smyčky. Ten umožňuje převést proudový vstup 0-20 mA nebo 4-20 mA na napětí. Převodník obsahuje celkem 5 svorek, 2 slouží k připojení proudové smyčky, 2 k napájení a 1 jako napěťový výstup. Napájení modulu je možné stejnosměrným napětím v rozsahu 7-35 V. Pro správné fungování převodníku musí být napájecí napětí vždy o něco vyšší než maximální požadované výstupní napětí. Pomocí dvou jumperů lze měnit výstupní napětí v rozsahu 0-2,5 V, 0-3,3 V, 0-5 V a 0-10 V. Ke přesné kalibraci nulového a maximálního výstupu slouží dvojice potenciometrů.

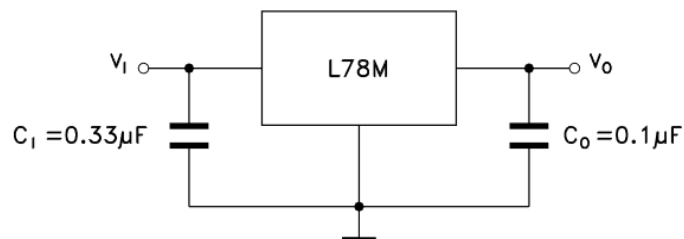
Na obrázku lze vidět zapojení 3 analogových vstupů skrze převodníky proudové smyčky. Analogové vstupy jsou značeny jako číslo CANON konektoru, číslo jeho pinu a názvem signálu, který je tímto způsobem realizován (např. P1.13 Mixer Reference). Výstupy z převodníků jsou zpracovány analogovými vstupy mikrokontroléru A0-A2.



Obr. 4-21 – Zapojení převodníku proudové smyčky

4.5 Zapojení analogových výstupů

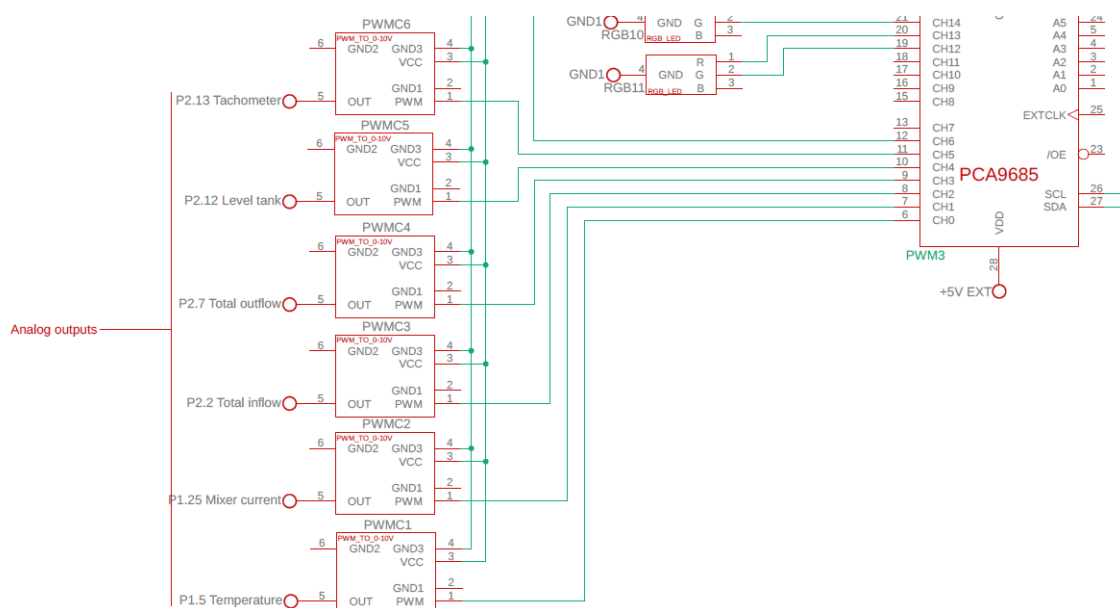
Analogový výstup simulátoru uskutečňuje převodník PWM signálu na napětí. Modul obsahuje 6 svorek, 2 slouží k napájení, 2 ke vstupnímu PWM signálu a 2 pro napěťový výstup. Vstupní napájení modulu je v rozsahu 12-30 V. Stabilní napájecí napětí modulu zajišťuje 3 pinový regulátor typu L78M12. Nejčastější praktické zapojení regulátoru je s kondenzátory mezi napěťovými piny a zemí (viz Obr. 4-22).



Obr. 4-22 – Zapojení regulátoru typu L78M [24]

PWM vstup je galvanicky oddělen od výstupu pomocí optočlenu. Amplituda PWM signálu může být v rozsahu 4,5-24 V. Pro správné fungování modulu je nutné připojit pomocí jumperu správný předřadný odpor optočlenu. O převod PWM signálu na výstupní napětí 0-10 V se stará čip LM138 s duálním operačním zesilovačem. Pro přesnou kalibraci výstupního napětí slouží potenciometr.

Na Obr. 4-23 je zobrazeno zapojení všech 6 analogových výstupů pomocí převodníků PWM signálů. Vstupní PWM signál je generován pomocí jednotlivých kanálů PWM expanderu (PCA9685), který je k Arduino připojen skrze I2C rozhraní. Analogové výstupy – vstupy do převodníku jsou značeny jako číslo CANON konektoru, číslo jeho pinu a názvem signálu, který je tímto způsobem realizován (např. P2.13 Tachometer).



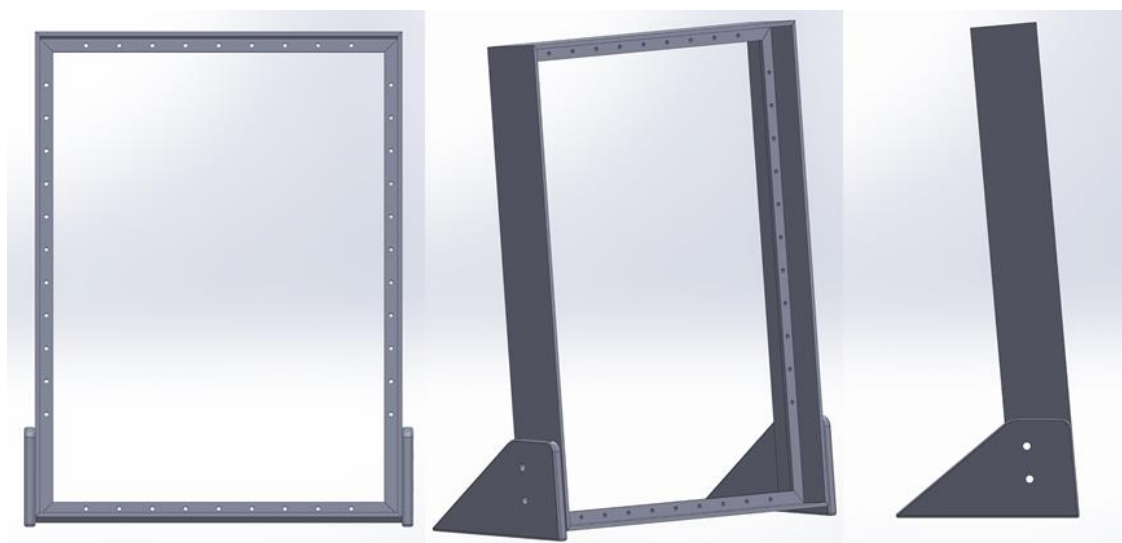
Obr. 4-23 – Zapojení převodníku PWM signálu

5. REALIZACE MECHANICKÉ KONSTRUKCE SIMULÁTORU

Požadavek na mechanickou konstrukci je především aby byla co nejlehčí. Se simulátorem bude často manipulováno a nesmí tedy být těžký. Zároveň je kladen důraz na reprezentativní vzhled, především ze přední strany. Konstrukce musí být dostatečně stabilní, aby se simulátor při ovládaní tlačítek uživatelem nepřevrátil. Realizace zapojení elektroniky simulátoru musí být přehledné. Z důvodu velkého množství vodičů je nutné vodiče vázat do svazků, nebo je schovat do kabelových lišt či jiných vhodných elektroinstalačních materiálů.

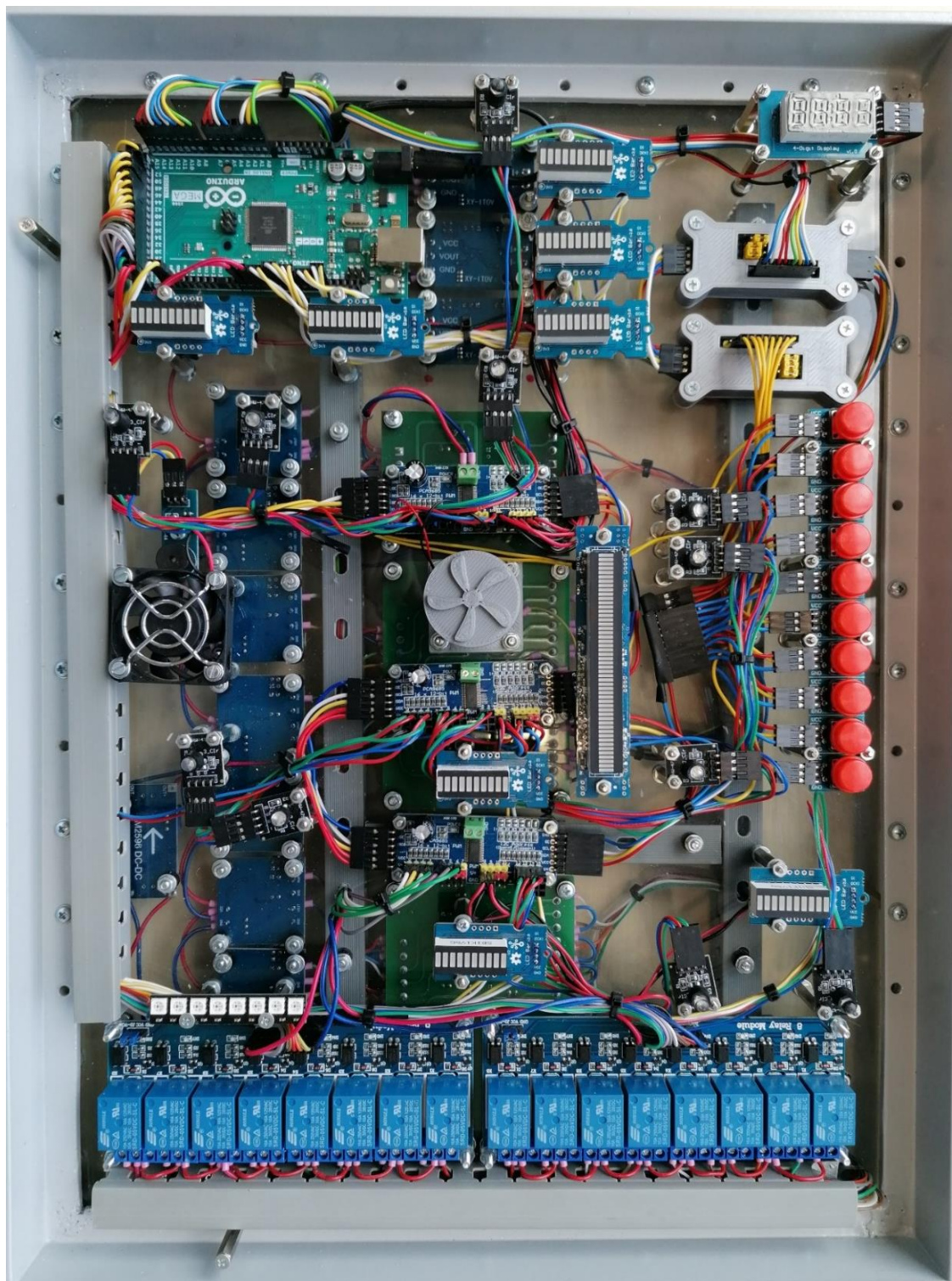
5.1 Reálná verze simulátoru

Konstrukce rámečku byla vyrobena z hliníkového T profilu, který byl nařezán pod úhlem 45° a svařen. Do vnitřní strany rámečku byly vyvrtány díry pro připevnění montážní desky s elektronikou a oba vnější panely. Z boku rámečku byly připevněny dvě hliníkové podpěry, které umožnily sklon simulátoru pod úhlem 5°. Tímto bylo dosaženo k pohodlnému ovládaní simulátoru skrze tlačítka. Podpěry dále zajišťují stabilitu celého zařízení.



Obr. 5-1 – Konstrukce rámečku

Veškerá elektronika simulátoru byla připevněna k čiré polykarbonátové desce pomocí šroubků, podložek a matek. Na některé moduly je kladen požadavek, aby byly ve stejné úrovni, z důvodu přesného lícování s předním panelem. Jedná o indikační moduly (RGB LED, bargrafy atd.). Tyto moduly byly proto ustaveny na vhodně dlouhých distančních sloupcích.

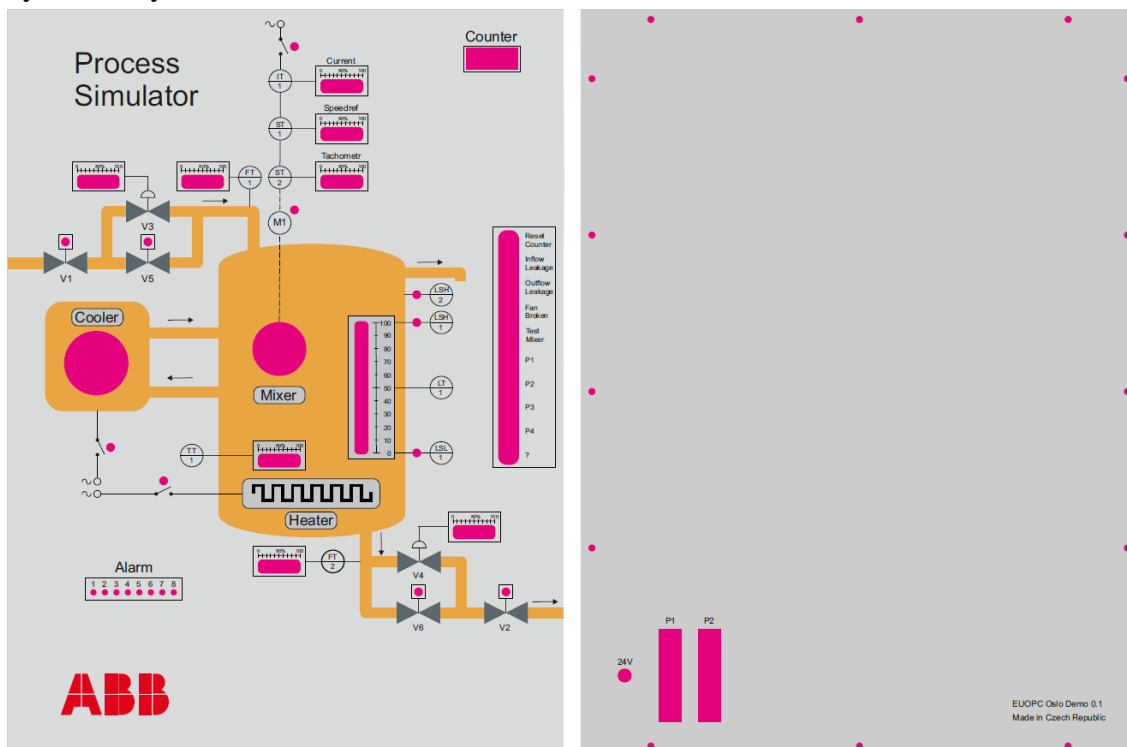


Obr. 5-2 – Reálné zapojení přední strany simulátoru

Montážní deska pro upevnění modulů je antistatická, což zajišťuje, že citlivá elektronika nebude zničena statickým nábojem, který může vzniknout při montáži. Vodiče jsou v maximální míře vedeny v kabelových žlabech. Pokud to nebylo možné, byly vodiče svázané do svazků. Toto opatření značně zpřehlednilo výsledné zapojení. Všechny vodiče byly opatřeny buď zakončovací dutinkou, nebo odnímatelným KONPC konektorem.

Tímto byla zajištěna jednoduchá montáž či demontáž při potenciální opravě zařízení. Jelikož je navíc celé zařízení modulární, bude oprava či výměna modulu velice snadná.

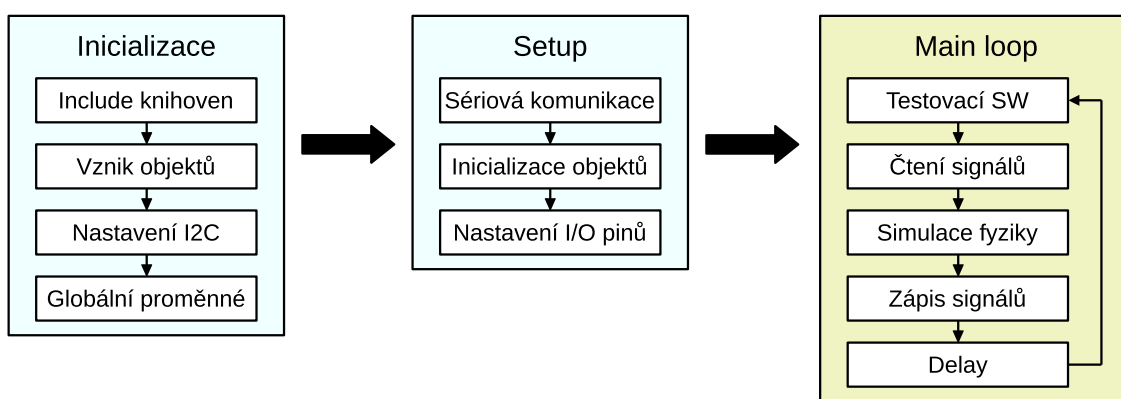
Oba vnější panely byly vyrobeny z eloxovaného hliníkového plechu, který je odolný vůči mechanickému poškození. Tyto plechy byly barevně potištěny a byly zde vyfrézovány otvory pro konektory a indikační moduly. Na Obr. 5-3 lze vidět návrh potisku plechů. Růžovou barvou jsou vyznačeny otvory, které byly do plechů vyfrézovány.



Obr. 5-3 – Vnější panely

6. IMPLEMENTACE ŘÍDICÍHO SOFTWARE SIMULÁTORU

Kapitola popisuje řídicí software simulátoru a jeho implementaci. Detailněji se zabývá popisem algoritmu pro simulaci fyziky, který byl zmíněn už v kapitole 3.1. Rovněž popisuje funkci testovacího kódu, jakožto diagnostického nástroje pro detekci chyb v zapojení řídicí elektroniky. Popisuje také proces čtení a zápisu signálů a použití příslušných knihoven. Pro vývoj softwaru bylo použito vývojové prostředí Arduino IDE a programovací jazyk Wiring, respektive C++. Řídicí SW lze rozdělit na tři základní celky, tedy inicializaci, setup a main loop (viz Obr. 6-1).



Obr. 6-1 – Blokové schéma řídicího SW

Inicializace zahrnuje připojení všech potřebných knihoven k projektu. K tomuto účelu slouží hlavní knihovna *OsloDemo.h*, jejíž další vlastnosti a funkce budou v textu uvedeny později. Součástí inicializace je také vytvoření objektů pro moduly uvedeny v kapitole 4.1. Objekt modulu slouží především pro počáteční nastavení komunikace (zapsání adres zařízení komunikujících skrze I2C) nebo pro volání požadovaných funkcí modulu. Dále jsou deklarovány a implicitně inicializovány globální proměnné, které se používají pro čtení a zápis signálů (viz Tabulka 2-1 a Tabulka 2-2).

Setup zajišťuje nastavení parametrů, které je nutné provést právě jednou, před spuštěním hlavní smyčky – main loop. Jedná se o nastavení komunikace přes sériovou linku a dále o počáteční inicializaci vytvořených objektů (vynulování počátečních stavů, nastavení frekvencí, zahájení komunikace). V rámci setupu se rovněž nastaví I/O piny mikrokontroléru a jejich počáteční hodnoty (INPUT/OUTPUT, HIGH/LOW).

Hlavní smyčku tvoří testovací kód, který je detailně popsán v kapitole 6.1. Main loop dále obsahuje funkce pro čtení a zápis signálů, které přímo komunikují s PLC. Součástí funkce pro zápis je samozřejmě i kód realizující obsluhu indikačních periférií. Výstupní signály jsou tedy zobrazovány na předním panelu simulátoru. Simulace fyziky čerpá ve velké míře z virtuálního návrhu simulátoru (viz podkapitola 3.1). Fungování fyziky bylo

úspěšně otestováno u virtuálního simulátoru a finální kód realizující simulaci je v podstatě již funkční algoritmus upravený na požadovanou syntaxi.

Program v main loop se vlastně chová jako PLC cyklus [25], kdy si přečte vstupy (čtení signálů), provede požadovaný program (simulace fyziky) a aktualizuje výstupy (zápis signálů). Časová prodleva mezi jednotlivými cykly je realizována pomocí zpoždění (delay), které lze v kódu nastavit pomocí konfigurační aplikace. U PLC se hodnota zpoždění, respektive skenovací periody, pohybuje řádově v desítkách ms [25].

6.1 Testovací kód

Vzhledem k množství použitých elektronických modulů, bylo potřeba vytvořit kód, který by sloužil k verifikaci jejich správného fungování. Skutečnost, že veškerá elektronika je přímo řízena mikokontrolérem, umožňuje otestovat správné zapojení a funkčnost všech modulů. Simulátor je tedy možné sestavit (viz Obr. 5-2) a finální zapojení je možné zkontrolovat pomocí testovacího kódu. V případě neúspěšného testu je odhalena chyba, kterou je pak možné přesně diagnostikovat a opravit. Tímto způsobem je spolehlivě ověřeno zapojení řídicí elektroniky a simulátor je připraven na implementaci řídicího SW. Testovací kód je tedy užitečným diagnostickým nástrojem pro detekci chyb v zapojení řídicí elektroniky, což je nezbytné pro prvotní oživování simulátoru nebo jeho případné opravy.

Testování funkce:

seg_test()

Inicializuje sedmisegmentový displej (nastaví požadovaný jas a vynuluje ho). Ověření funkčnosti displeje je realizováno inkrementací displeje s krokem jedna od nuly do deseti. Je zde využito funkcí knihovny *TM1637Display.h*.

motor_fan_test()

Testuje funkčnost DC motorku a ventilátoru. Roztáčení motorku probíhá pozvolna z nízkých otáček do vysokých, aby se zabránilo jeho mechanickému opotřebení. Test rovněž ověřuje funkčnost akustického bzučáku.

relay_test()

Postupně aktivuje každý kanál relé modulu se sekundovou prodlevou (půl sekundy zapnuto, půl sekundy vypnuto). Tímto způsobem je ověřeno fungování digitálních výstupů simulátoru.

rgb_test()

Testuje správnou funkčnost a zapojení RGB LED modulů. Postupně je aktivován jeden RGB modul za druhým (aktivace je indikována červenou nebo zelenou barvou). Tímto je současně docíleno i otestování správného fungování PWM expandérů, které přímo spínají RGB LED moduly. Je zde využito funkcí knihovny *Adafruit_PWMServoDriver.h*.

bg_test()

Ověřuje správné fungování a zapojení bargraf modulů, které jsou použity pro zobrazení analogového signálu. Najednou inkrementuje hodnotu s prodlevou 0,1 sekundy. Výsledkem je plynulý přechod z 0 % do 100 % u všech bargraf modulů. Je zde využito funkcí knihovny *Grove_LEDBar.h*.

led_panel_test()

Inicializuje panel s osmi RGB LED diodami (nastaví požadovaný jas a barvu diod). Podobně jako u bargraf modulů je funkčnost a zapojení modulu otestována postupnou inkrementací s časovou prodlevou. Je zde využito funkcí knihovny *Adafruit_NeoPixel.h*.

6.2 Simulace fyziky

Jak již bylo uvedeno výše v textu, celý program v hlavní smyčce se chová jako PLC cyklus. Proto i algoritmus simulace fyziky musí být navržen cyklickým způsobem (simulované signály se budou měnit s požadovanou dobou zpoždění – taskem). U digitálních signálů, které jsou například zpětná odezva ventilu nebo plováčkového snímače hladiny, se simulovaný signál zapíše vždy při splnění podmínky v daném tasku. Splněnou podmínkou se myslí například dosažení hladiny v nádrži, která sepne plováčkový snímač. U analogových signálů, které jsou například teplota v nádrži, nebo množství kapaliny v nádrži, je nutné vždy spočítat přírůstek veličiny za jeden task. S každou periodou tasku se přírůstek přičte k předchozí hodnotě (respektive odečte, pokud se veličina zmenšuje).

Simulace změny objemu kapaliny v nádrži

Objem kapaliny v nádrži se mění při přitékání a odtékání kapaliny. Tuto skutečnost popisuje následující rovnice.

$$LV(t + 1) = LV(t) + I_{PT} - O_{PT} \quad (6.1)$$

Kde LV (liquid volume) představuje objem kapaliny v nádrži [l], I_{PT} (inflow per task) přítok za jeden task [$l.task^{-1}$] a O_{PT} (outflow per task) odtok za jeden task [$l.task^{-1}$].

Je zřejmé, že přítok (I) do nádrže není konstantní, ale bude se měnit v závislosti na stavech ventilů ve vstupním potrubí. Tuto závislost lze vyjádřit rovnicí 6.2 uvedenou níže (rovnice vychází ze schématu napouštění viz Obr. 2-2).

$$I \in \langle 0, Q_{max} \rangle$$
$$I = V_1 \left[\left(\frac{V_3}{100} \right) \cdot Q_{max} + (V_5 \cdot Q_{BPmax} \cdot Q_{max}) \right] + L \quad (6.2)$$

Kde I je přítok [$m^3.h^{-1}$], Q_{max} maximální objemový průtok vstupního potrubí [$m^3.h^{-1}$], V_1 a V_5 stavy digitálních ventilů [*on/off*], V_3 stav lineárního ventilu [%], Q_{BPmax} maximální objemový průtok bypass ventilu a L konstanta úniku v potrubí [$m^3.h^{-1}$]. Hodnota přítoku může být v rozmezí od nuly do maximálního objemového průtoku

vstupního potrubí – Q_{max} . Signál V_1 představuje stav hlavního ventilu V_1 (1 – zapnuto, 0 – vypnuto). Je-li ventil uzavřen, je uzavřeno celé vstupní potrubí, a tedy jím žádná kapalina neprotéká. V opačném případě má přítok nenulovou hodnotu a záleží na stavech řídicího ventilu V_3 (V_3 – reprezentuje lineární rozsah ventilu v procentech 0-100 %) a bypass ventilu V_5 , který má omezený objemový průtok – Q_{BPmax} . Netěsnosti potrubí a potenciální úniky kapaliny reprezentuje konstanta L .

$$I_{PT} = I \frac{task}{3600} \quad (6.3)$$

Výpočet přítoku za jeden task, který lze dosadit do rovnice 6.1 zobrazuje rovnice 6.3, kde I_{PT} je přítok za jeden task [$l.task^{-1}$], $task$ hodnota tasku [ms]. Číselná konstanta 3600 zajišťuje převod z $m^3.h^{-1}$ na $l.task^{-1}$.

Obdobným způsobem lze vypočítat i odtok za jeden task, který se rovněž dosadí do rovnice 6.1.

$$O \in \langle 0, Q_{max} \rangle$$

$$O = V_2 \left[\left(\frac{V_4}{100} \right) \cdot Q_{max} + (V_6 \cdot Q_{BPmax} \cdot Q_{max}) \right] + L \quad (6.4)$$

$$O_{PT} = O \frac{task}{3600} \quad (6.5)$$

Kde O je odtok [$m^3.h^{-1}$], Q_{max} maximální objemový průtok výstupního potrubí [$m^3.h^{-1}$], V_2 a V_6 stavy digitálních ventilů [on/off], V_4 stav lineárního ventilu [%], Q_{BPmax} maximální objemový průtok bypass ventilu a L konstanta úniku v potrubí [$m^3.h^{-1}$].

Simulace změny teploty

Jak již bylo v textu zmíněno, změna teploty může nastat ve třech případech. První z případů je cílena regulace teploty kapaliny za užitím chladiče nebo topného tělesa. Množství kapaliny je tedy ochlazováno, nebo naopak ohříváno. Druhý případ nastává, když do nádrže s kapalinou o určité teplotě přitéká kapalina s jinou teplotou. Dochází tedy k mísení kapalin a „vyrovnání“ jejich teplot na teplotu novou. Změnu teploty také ovlivňuje únik tepla do okolí. Ohřátá kapalina, jejíž teplota je vyšší než teplota okolí, se v čase samovolně ochlazuje, skrze stěny nádrže. Změna teploty kapaliny v nádrži lze popsat jednoduchou rovnicí 6.6, uvedenou níže.

$$T(t + 1) = T(t) + T_{PT} \quad (6.6)$$

Kritické je pro simulaci teploty vypočtení hodnoty T_{PT} (temperature per task). Hodnota T_{PT} představuje změnu teploty ve stupních za jeden task [$^{\circ}C/task$]. Její vypočtení se liší pro každý se tří možných případů.

V prvním případě je změna teploty způsobena regulací za užitím topného tělesa. Zahřátím topné spirály, nebo jiného zařízení plnící funkci ohříváče, dochází k přenosu tepla do kapaliny. Proces změny teploty v tomto případě popisuje diferenciální rovnice 6.7.

$$\frac{m_H c_H m_L c_L}{K_H} \frac{d^2 \vartheta}{dt^2} + \left[m_H c_H \left(\frac{K_L}{K_H} + 1 \right) + m_L c_L \right] \frac{d\vartheta}{dt} + K_L \vartheta = p(t) \quad [26](6.7)$$

Kde m_x je hmotnost $[kg]$, c_x je měrná tepelná kapacita $[W.s. kg^{-1}.^{\circ}C^{-1}]$, K_x je koeficient přestupu tepla mezi plochou dvou materiálů $[W.^{\circ}C^{-1}]$, ϑ $[^{\circ}C]$ vyjadřuje rozdíl teploty vody a teploty okolí ($\vartheta = \vartheta_L - \vartheta_A$) a $p(t)$ představuje topný výkon $[W]$. Index H určuje veličiny a konstanty patřící k topnému tělesu (heater) a index L k ohřívané kapalině (liquid). Rovnici tedy představuje diferenciální rovnici druhého řádu, kterou je možné přepsat následujícím způsobem.

$$a_2 \frac{d^2 \vartheta}{dt^2} + a_1 \frac{d\vartheta}{dt} + a_0 \vartheta = p(t) \quad (6.8)$$

Kde platí že:

$$\begin{aligned} \vartheta(0) &= 0 & \frac{d\vartheta(0)}{dt} &= 0 \\ a_2 &= \frac{m_H c_H m_L c_L}{K_H} & a_1 &= \left[m_H c_H \left(\frac{K_L}{K_H} + 1 \right) + m_L c_L \right] & a_0 &= K_L \end{aligned}$$

Je-li na rovnici 6.8 použita Laplaceova transformace (LT), bude výsledný obraz roven:

$$\Theta = \frac{P}{p(a_2 p^2 + a_1 p + a_0)} \quad (6.9)$$

Pro vyřešení diferenciální rovnice je nutné použít inverzní Laplaceovu transformaci. Řešení tímto způsobem ovšem není úplně triviální. Rovnici by sice bylo možné předpočítat ručně, nebo v některém z výpočetních programů (Matlab, WolframAlpha...), nicméně tímto by bylo docíleno pouze vyřešení rovnice pro aktuální parametry. Koeficienty a_2 , a_1 a a_0 nejsou konstantami, ale obsahují veličinu m_L , tedy hmotnost kapaliny, která se v čase může měnit. V případě, že do nádrže bude přitékat kapalina a zároveň bude probíhat ohřev, bude docházet každý task ke změnám těchto koeficientů. Diferenciální rovnice se tedy bude v čase měnit a bude mít různá řešení. Inverzní laplaceovu transformaci bude v tomto případě nutné počítat v každém tasku. Toto je s použitím mikrokontroléru dosti výpočetně náročné. Pro řešení diferenciální rovnice byl proto zvolen následující postup.

Diferenciální rovnice sice v čase mění své parametry, ovšem její tvar zůstává neměnný, vždy bude řešena diferenciální rovnice druhého řádu. Je zřejmé, že koeficienty a_2 , a_1 a a_0 budou vždy kladné (záporná hmotnost a tepelné konstanty nedávají z hlediska fyziky smysl). Stejně tak výkon $p(t)$ bude kladný a navíc konstantní (z tohoto důvodu byl v obrazu LT transformován jako konstanta P). Obraz LT (rovnice 6.9) lze rozložit v obecném tvaru na součet parciálních zlomků.

$$\Theta = \frac{P}{p(a_2 p^2 + a_1 p + a_0)} = \frac{P}{p(p+a)(p+b)} = \frac{A}{p} + \frac{B}{p+a} + \frac{C}{p+b}$$

Kde:

$$A = \frac{P}{a_0} \quad B = -\frac{P}{a_0} - C \quad C = \frac{P(b \cdot a_2 - a_1)}{a_0 \cdot a_2(a-b)}$$

Vzhledem k tomu, že je obraz LT rozložen na součet parciálních zlomků, lze pro jeho vyřešení použít známé vzorce zpětné LT. Obecné řešení diferenciální rovnice 6.8 je tedy následující:

$$\vartheta(t) = A + B e^{-at} + C e^{-bt} \quad (6.10)$$

Obecné řešení rovnice obsahuje pouze exponenciální funkce a není problém jej v mikrokontroléru implementovat. Výsledná funkce popisuje změnu teploty kapaliny, vzhledem k teplotě okolí, která je pro jednoduchost neměnná $\vartheta(t) = \vartheta_L(t) - \vartheta_A$. Podobným způsobem lze realizovat i proces ochlazování kapaliny chladičem. Tento proces bude popisovat stejná diferenciální rovnice, ve které však bude topný výkon $p(t)$ záporný.

Hodnota T_{PT} je tedy v tomto případě vypočtena jako součet tepelných účinků chladiče T_{PTC} a topného tělesa T_{PTH} . Tepelné účinky zařízení se rovnají hodnotě výsledné funkce 6.10 podělené taskem.

$$T_{PTC} = \frac{\vartheta_C}{task} \quad T_{PTH} = \frac{\vartheta_H}{task} \quad T_{PT} = T_{PTC} + T_{PTH} \quad (6.11)$$

Výše uvedený výpočet teploty sice počítá s proměnlivým množstvím kapaliny v nádrži. Nelze ovšem zanedbat efekt mísení kapalin dvou různých teplot, ke kterému dochází v případě, kdy kapalina v nádrži je zahřátá na určitou teplotu a je otevřené vstupní potrubí. Tento druhý případ popisuje kalorimetrická rovnice přepsána do níže uvedené podoby. Vychází se z rovnice 2.1 uvedené v kapitole 2.1. Měrná tepelná kapacita je u obou mísících se kapalin stejná, lze ji tedy zkrátit. Stejně tak hustota kapalin je shodná, lze tedy počítat pouze s objemem kapalin. Výsledkem je vyjádření výsledné teploty mísících se kapalin následující rovnicí.

$$T(t+1) = \frac{I_{PT} \cdot T_i + LV \cdot T(t)}{I_{PT} + LV} \quad (6.12)$$

Kde T je teplota [$^{\circ}C$], T_i teplota přitékající kapaliny [$^{\circ}C$], LV (liquid volume) objem kapaliny v nádrži [l], I_{PT} (inflow per task) přítok za jeden task [$l \cdot task^{-1}$].

Ke třetímu případu dochází, když je teplota kapaliny vyšší než teplota okolí. V tomto případě dochází k únikům tepla, skrze stěny nádrže. Množství tepla, které se z nádrže ztrácí je úměrné koeficientu přestupu tepla vynásobeného rozdílem teplot:

$$Q_{OUT} \sim K_L(\vartheta_L(t) - \vartheta_A) \quad [26](6.13)$$

Množství kapaliny představuje akumulátor energie, tuto úvahu potvrzuje i přítomnost astatismu prvního řádu v obrazu LT 6.9. Proces ohřevu tedy bude způsobovat akumulaci energie ve formě tepla. Množství akumulovaného tepla se dá spočítat jako:

$$Q_{SUM} = c_L \cdot m_L \cdot \Delta T \quad [27](6.14)$$

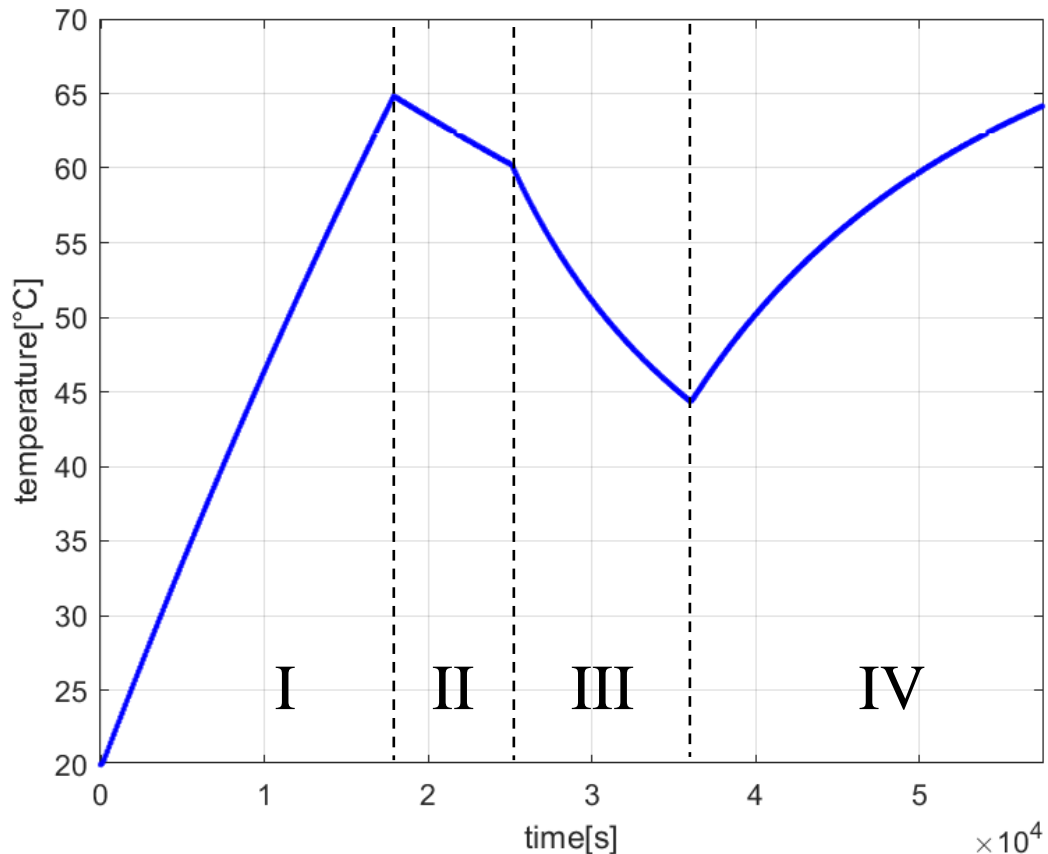
Kde Q_{SUM} představuje množství akumulovaného tepla v kapalině $[J]$ a ΔT představuje rozdíl teplot kapaliny a okolí $[^{\circ}C]$. Ve třetím případě tedy lze simulaci změny teploty kapaliny realizovat rovnicí:

$$T_L(t) - T_A = \frac{Q_{SUM}(t) - Q_{OUT}(t)}{m_L \cdot c_L} \quad (6.15)$$

Kde T_L je teplota kapaliny $[^{\circ}C]$, T_A teplota okolí $[^{\circ}C]$, Q_{SUM} množství akumulovaného tepla v kapalině, Q_{OUT} množství tepla, které z nádrže uniká. Změna T_{PT} je pak vypočtena rozdílem teplot poděleným taskem.

$$T_{PT} = \frac{T_L - T_A}{task} \quad (6.16)$$

Všechny tři případy zobrazuje graf níže (Obr. 6-2), který je rozdělen do sekcí o časovém intervalu uvedeném v hodinách. Sekce I představuje fázi ohřevu (5 h), II fázi samovolného ochlazování (2 h), III přítok chladnější kapaliny (3 h) a IV současný ohřev a přítok chladnější kapaliny (6 h). Simulace je provedena pro 1000 l vody o počáteční teplotě $20^{\circ}C$, výkon topného tělesa 12 kW a konstantní teplotu okolí $20^{\circ}C$ (ostatní proměnné a koeficienty jsou definovány v matlabovém skriptu v přílohách práce)

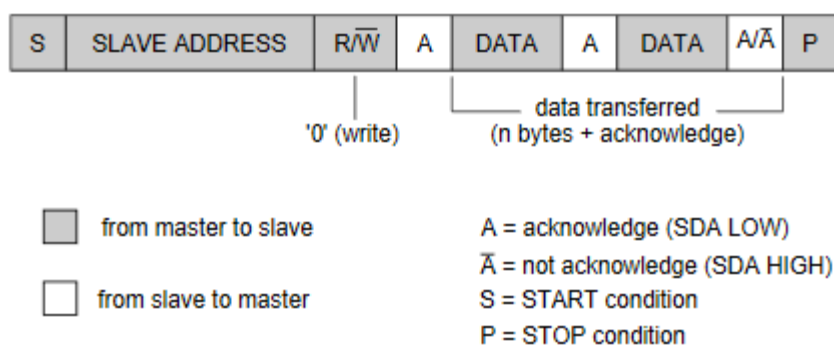


Obr. 6-2 - Simulace teploty v Matlabu

6.3 Čtení a zápis signálů

Čtení signálů je proces, kdy simulátor získává aktuální hodnoty z kanálů výstupních karet PLC a uchovává si je pro další výpočty. Pro rekapitulaci – spojení simulátoru s PLC je řešeno kabely s CANON konektory a rozvodovou svorkovnicí. Tímto způsobem je docíleno, že jeden vodič slouží právě pro přenos jednoho signálu. O převod úrovně signálů se starají převodníky. Teoreticky tedy stačí, když je výstup z převodníku přiveden na pin mikrokontroléru, který je konfigurovaný pro čtení. Funkcí Arduina – *digitalRead(pin)* nebo *analogRead(pin)* lze hodnotu přečíst a uložit do paměti. Takto je to proveditelné a čtení je v simulátoru tímto způsobem opravdu realizováno, ovšem mikrokontrolér nedisponuje dostatečným počtem konfigurovatelných I/O pinů. Z tohoto důvodu je použit I/O expander modul. Ten se simulátorem komunikuje skrze I2C sběrnici.

I2C je oblíbená synchronní sériová sběrnice, která pomocí dvou vodičů SDA (Serial Data) a SCL (Serial Clock) umožňuje komunikaci MultiMaster-MultiSlave. Přenos dat je zobrazen na Obr. 6-3 níže. Data jsou posílány v osmibitových balíčcích, které rozdělují tzv. acknowledge bity – A. Ty slouží k verifikaci, že přenos předchozího bytu byl úspěšný. Komunikaci vždy začíná master (mikrokontrolér) za pomoci START bitu – S. Dále následuje osm bitů, které obsahují sedmibitovou adresu zařízení, se kterým chce master komunikovat – Slave address a informaci o tom, zda je požadavek data číst, nebo zapisovat – R/W' bit. Přenos může obsahovat libovolný počet bytů, jediná podmínka je, že za každým bytem musí následovat A bit. Přenos ukončuje master STOP bitem P [28].



Obr. 6-3 Master-vysílač adresuje slave zařízení-přijímač [28]

Čtení signálů pomocí I/O expander modulu probíhá následujícím způsobem. V inicializační sekci řídicího softwaru vznikne objekt, který se váže ke konkrétnímu modulu. Dále je objekt inicializován a je mu přiřazena unikátní sedmibitová adresa. Stejná adresa je nastavena i fyzicky na modulu pomocí adresovacích jumperů. V setup sekci se objekt nastaví tak, aby byl připraven na komunikaci s mikrokontrolérem. Pokud je v hlavní smyčce, pomocí objektu modulu zavolána metoda *read(pin)*, Arduino (jako master) zahájí přenos a přečte hodnotu pinu I/O expanderu (slave). Jsou-li data úspěšně přečtena, přenos se ukončí a hodnota se zapíše do příslušné proměnné.

Přečtená data jsou zpracována a na základě simulačních výpočtů jsou generovány výstupní signály.

Zápis signálu je proces, při kterém simulátor výstupní signály posílá na kanály vstupních karet PLC. Realizace spojení mikrokontroléru s PLC a převodu úrovní signálů je stejná jako u čtení. Pro zapsání signálu na požadovaný pin Arduina jsou použity funkce *digitalWrite(pin)* a *analogWrite(pin)*. Z důvodu nedostatku výstupních PWM pinů Arduina jsou použity PWM expander moduly. S mikrokontrolérem komunikují opět skrze I2C sběrnici. Zápis signálů pomocí modulů probíhá podobným způsobem jako u čtení. Vytvoří se objekt, kterému se přiřadí stejná adresa jako fyzickému modulu (v tomto případě je adresa modulu nastavena spájením příslušných plošek). V setup sekci dojde k nastavení frekvence PWM a připravení modulu ke komunikaci s mikrokontrolérem. Pro zápis hodnoty PWM modulu na příslušný pin je použito volání metody *setPWM(pin, on, off)*.

6.4 Knihovny

Funkcí hlavní knihovny je definování konstant, kterými lze nastavit různé parametry simulátoru. Obsahuje definice analogových a digitálních pinů mikrokontroléru, ve tvaru: `#define pojmenování_pinu číslo_pinu`. Ve své původní verzi knihovna obsahovala také konstanty, kterými lze parametrizovat simulátor (např. nastavení velikosti nádrže, maximální teploty atd.). Funkce parametrizace simulátoru byla postupně převedena na konfigurační aplikaci, o které hovoří kapitola *Konfigurační rozhraní simulátoru*. Hlavní knihovna *OsloDemo.h* také includuje všechny ostatní knihovny (viz Obr. 6-4 Struktura knihoven). Knihovna *OsloDemo.h* strukturuje a zpřehledňuje kód, a tedy činí projekt čitelnější.

Funkce knihoven:

Arduino.h [29]

Základní knihovna pro programování Arduina. Obsahuje definice konstant a základních matematických funkcí (např. *max()*, *min()* atd.). V programu se využívá především funkcí pro zápis a čtení hodnot pinů mikrokontroléru (*digitalRead/Write()*, *analogRead/Write()*)

Wire.h [30]

Knihovna umožňuje komunikaci Arduina s I2C zařízeními. Využívají jí tedy všechny níže uvedené knihovny.

TM1637Display.h [31]

Knihovna, která zajišťuje funkce potřebné pro ovládání sedmi segmentového displeje. Využívá se funkcí pro zahájení komunikace, nastavení jasu, mazání a zobrazování číslic.

Grove_LED_Bar.h [32]

Umožňuje komunikaci a ovládání bargraf modulů. Obsahuje funkce, kterými lze, měnit jas, barvu LED diod. Také lze spínat jednotlivé diody nebo nastavit požadovanou úroveň,

kterou má bargraf modul zobrazit.

PCF8574.h [33]

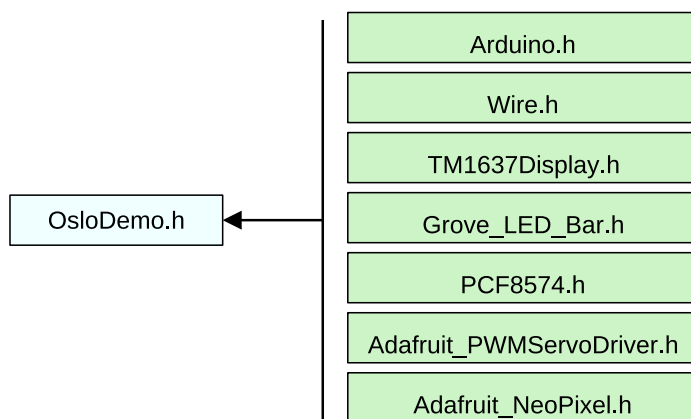
Knihovna se stará o spolehlivou komunikaci Arduina s I/O expander modulem. V hlavním kódu je především využito funkcí pro zápis a čtení dat pinů expander modulu.

Adafruit_PWMServoDriver.h [34]

Knihovna, která zajišťuje ovládání PWM expander modulů. Obsahuje například funkce pro nastavení komunikace s modulem, frekvence PWM a zápisu signálu na daný pin modulu.

Adafruit_NeoPixel.h [35]

Umožňuje ovládání LED panelu typu NeoPixel. Pomocí knihovnických funkcí lze nastavit jas a barvu diody pomocí RGB kódu. Rovněž umožňuje rozsvěcovat a zhasínat jednotlivé diody. LED panelu



Obr. 6-4 Struktura knihoven

7. KONFIGURAČNÍ ROZHRAŇÍ SIMULÁTORU

Řešení simulátoru je obecné, změnou příslušných proměnných a konstant v kódu lze tedy simulátor libovolně měnit (měnit velikost nádrže, maximální průtok, počáteční teplotu kapaliny atd.). Tato akce však vyžaduje základní znalosti programování a porozumění kódu. Za účelem jednoduché parametrizace simulátoru i pro koncového uživatele bylo potřeba vytvořit konfigurační rozhraní. Jako toto rozhraní slouží aplikace, díky které lze simulátoru měnit jeho parametry. Změnou parametrů simulátoru lze měnit soustavu, ve které probíhá technologický proces (více o parametrech je napsáno v kapitole 7.3). Konfigurační rozhraní je realizováno formulářovou aplikací spustitelnou na počítači s operačním systémem Windows. Aplikace je napsána v jazyce C# s využitím vývojového prostředí Visual Studio a .NET framework. Aplikace umožňuje připojení osobního počítače k simulátoru a zapsání požadovaných parametrů skrze sériový port.

7.1 Realizace konfigurační aplikace

Komunikace mezi simulátorem a konfigurační aplikací je realizována přes sériový COM port. Jedná se o stejný COM port, který je použit pro nahrání programu do mikrokontroléru. Po restartu simulátoru je na sériový monitor zaslána otázka, zda chce uživatel konfigurovat simulátor. Simulátor je nastaven na defaultní parametry, pokud uživatel zvolí odpověď „ne“, nebo vyprší definovaný čas pro získání odpovědi. Je-li zvolena možnost „ano“, simulátor čeká na připojení konfigurační aplikace.

Pro posílání dat mezi aplikací a simulátorem byl vytvořen jednoduchý komunikační protokol. Pro oddělení jednotlivých zpráv od sebe byl zvolen počáteční a koncový znak. Jednotlivé byty jsou tedy načítány do řetězce, ze kterého je následně získána zpráva, na základě oddělovacích znaků. Zasílané zprávy jsou vždy ve tvaru [MESSAGE], kdy hranaté závorky představují oddělovací znaky.

Realizace přenosu dat z aplikace do simulátoru je následující. Konfigurační aplikace vyžaduje zvolení sériového portu, přes který se má připojit k simulátoru. Je-li zvolen správný sériový port, aplikace pošle výzvu simulátoru (PING), který na výzvu odpoví (PONG). Není-li odpověď simulátoru v požadované době doručena, zahlásí aplikace chybu v podobě odpojeného simulátoru. V případě, že simulátor odpoví, je uživateli umožněno zadat data. Zapsání dat z aplikace do simulátoru se realizuje klikem na tlačítko „write“. Po aktivaci tlačítka je uživatel vyzván, aby potvrdil zápis dat do simulátoru. Aplikace znovu pošle výzvu (PING), zda je simulátor připojen. Pokud ano, data se zapíší do simulátoru a konfigurační aplikace je ukončena. Program simulátoru začne vykonávat kód hlavní smyčky, kde již probíhá simulace fyziky. V případě, že simulátor nepošle odpověď (PONG), aplikace zahlásí chybu a pro nové zapsání parametrů je nutné ji restartovat. Proces přenosu dat z aplikace do simulátoru popisuje následující obrázek. Písmena a-e vyznačují jednotlivé sekce programu.

```

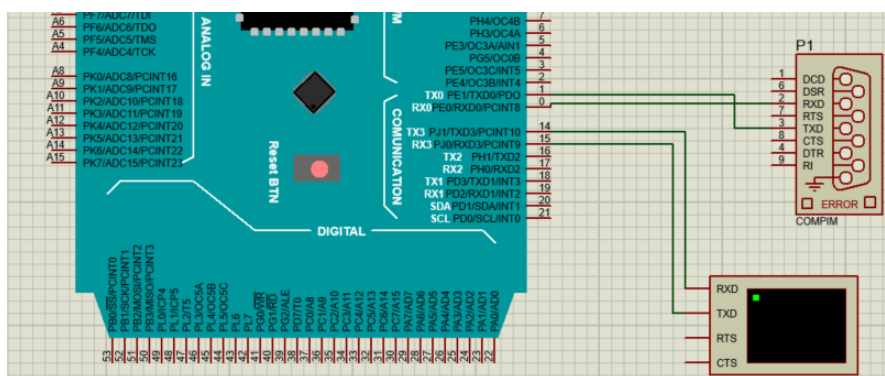
Virtual Terminal
a { Do you want to configure simulator? y/n
    Simulator configuration started. Open configuration application.
b { I received:
    [PING]
c { I received:
    [PING]
d { I received:
    [40;30;30;200;0;20;0;180;10000;10000;50]
    I received:
    [DISCONNECTED]
e { 0.40
    30.00
    30.00
    200.00
    0.00
    20.00
    0.00
    180.00
    10000.00
    10000.00
    50
  
```

Obr. 7-1 - Výpis ze sériového monitoru simulátoru

- a – Dotaz, zda chce uživatel konfigurovat simulátor
- b – Ověření připojení simulátoru
- c – Zpráva pro zápis dat
- d – Zpráva, že proběhl zápis dat a odpojení aplikace
- e – Kontrolní výpis přijatých dat

7.2 Demonstrace aplikace

Z důvodu virové epidemie Covid-19 nebylo možné aplikaci testovat na reálném simulátoru (zařízení bylo uchováno v ABB univerzitě, kde byl omezený přístup). Pro ověření funkčnosti aplikace bylo tedy využito vývojové prostředí Proteus, které se primárně využívá pro návrh elektronických obvodů a jejich simulaci.

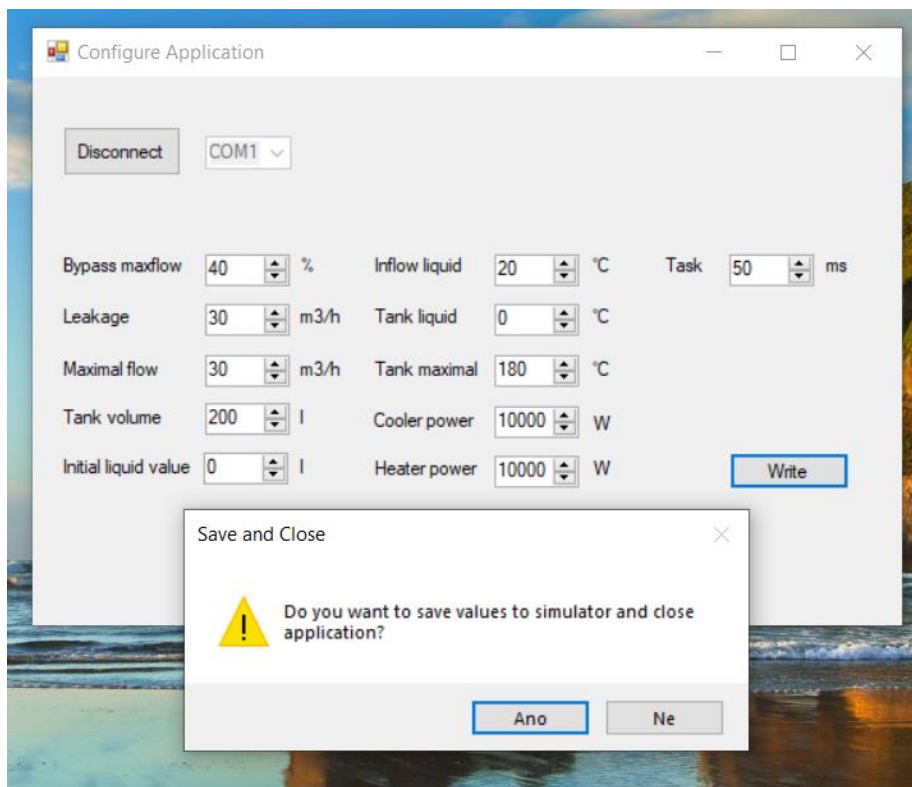


Obr. 7-2 - Realizace emulátoru Arduina v programu Proteus

S použitím Arduino knihovny určené pro Proteus bylo možné mikrokontrolér simulovat na osobním počítači. Do emulátoru Arduina byl nahrán stejný kód, jako do reálného simulátoru. Spojení mikrokontroléru a konfigurační aplikace, nebylo realizováno fyzicky USB kabelem, ale bylo nutné je připojit přes virtuální sériový port. K simulaci COM

portu byl použit software VSPE (Virtual Seriál Port Emulator). Jelikož se kód nenahrával přes vývojové prostředí Arduino IDE, které obsahuje i sériový monitor Arduina, bylo nutné připojit k emulátoru virtuální terminál. Realizace v prostředí Proteus je uvedena na Obr. 7-2.

Na takto připraveném emulátoru bylo možné ověřit funkčnost konfigurační aplikace. Aplikace je realizována jako spustitelný .exe soubor. Pro připojení aplikace k emulátoru je nutné zvolit sériový port pomocí rozbalovacího seznamu. Aplikace zobrazí všechny dostupné porty na daném počítači. Po zvolení správného portu (ten který je využíván pro komunikaci s emulátorem), se aplikace k simulátoru připojí aktivací tlačítka „Connect“. Při úspěšném připojení k simulátoru aplikace umožní konfiguraci parametrů simulátoru. Parametry se dají upravit přepsáním hodnoty v číselném poli, nebo pomocí inkrementačních šipek. V případě, že je uživatel spokojen se zadanými parametry, může pomocí tlačítka „Write“ zapsat data do simulátoru. Aplikace se zeptá uživatele, zda si přeje zapsat data a ukončit aplikaci. Po odsouhlasení jsou data zapsána do simulátoru a aplikace je ukončena.



Obr. 7-3 - Konfigurační aplikace

7.3 Parametry simulátoru

Změna parametrů simulátoru představuje změnu soustavy technologického procesu. Všechny konfigurovatelné parametry jsou zobrazeny na Obr. 7-3 - Konfigurační aplikace. První sloupec parametrů souvisí s kapalinou a jejich význam je následující:

- **Bypass maxflow** představuje maximální objemový průtok skrze bypass ventil v procentech (digitální ventily V5 a V6). Parametr tedy určuje omezení průtoku skrze bypass ventily. Výchozí hodnota je 40 % a rozsah pro konfiguraci parametru je 0-100 %.
- **Leakage** parametr reprezentuje konstantu netěsnosti ventilů. Definuje úniky vstupního a výstupního potrubí v m³/h. Výchozí hodnota je 30 m³/h a parametr může nabývat pouze kladných hodnot.
- **Maximal flow** parametr představuje maximální objemový průtok vstupního a výstupního potrubí v m³/h. Výchozí hodnota je 30 m³/h a parametr může nabývat pouze kladných hodnot.
- **Tank volume** představuje objem nádrže kapaliny v litrech. Výchozí hodnota je 200 l a parametr může nabývat pouze nezáporných hodnot.
- **Initial liquid volume** reprezentuje počáteční objem kapaliny v nádrži v litrech. Výchozí hodnota je nula a parametr může nabývat pouze nezáporných hodnot.

Druhý sloupec parametrů souvisí s teplotou a jejich význam je následující:

- **Inflow liquid (temperature)** představuje teplotu přitékající kapaliny ve stupních celsia. Výchozí hodnota je 20 °C a rozsah pro konfiguraci parametru je 0-50 °C.
- **Tank liquid (temperature)** je parametr reprezentující výchozí teplotu kapaliny v nádrži ve stupních celsia. Výchozí hodnota je 20 °C a parametr může nabývat pouze nezáporných hodnot.
- **Tank maximal (temperature)** reprezentuje maximální teplotu nádrže ve stupních celsia. Výchozí hodnota je 180 °C a rozsah pro konfiguraci parametru je 150-200 °C.
- **Cooler power** parametr definuje výkon chladiče ve wattech. Výchozí hodnota je 10 000 W a parametr může nabývat pouze nezáporných hodnot.
- **Heater power** parametr definuje výkon topného tělesa ve wattech. Výchozí hodnota je 10 000 W a parametr může nabývat pouze nezáporných hodnot.

Posledním parametrem je **Task**, kterým se nastavuje časová prodleva mezi jednotlivými cykly simulace v ms. Výchozí hodnota je 50 ms a rozsah pro konfiguraci parametru je 20-3000 ms.

8. ZÁVĚR

Výsledné zařízení bylo úspěšně otestováno a odladěno v prostorách ABB univerzity, kde bylo už několikrát využito pro školící účely. Prvotní ladění bylo realizováno forcováním dat na I/O kanály PLC systému s následným ověřením správných odezev simulátoru. V tomto kroku bylo spolehlivě ověřeno, že simulátor je schopen komunikovat s PLC systémem. V druhém kroku se ověřovalo správné naprogramování simulace fyziky. K tomuto účelu sloužila již naprogramovaná řídicí aplikace, která je součástí kurzu T315 na ABB systém 800xA. Spolehlivost simulace technologického procesu ověřil kolega, který daný kurz běžně školí. Jelikož kód čerpal z již odladěné verze virtuálního simulátoru, ověření simulace funkčnosti proběhlo úspěšně. Byl potřebný jen drobný zásah do simulace teploty, jelikož docházelo k chybám při přechodu ze stavu ohřevu do stavu chlazení a naopak. Chyba byla způsobena zanedbáním vlivu chladiče/ohříváče, který po nějakou dobu reguloval teplotu a byl vypnut. I po vypnutí daného zařízení v reálných podmínkách v praxi dochází k přenosu energie mezi zařízením a médiem v nádrži. Jinými slovy, i po vypnutí zařízení se po nějaký čas snižuje/zvyšuje teplota média.

Výsledný produkt splňuje požadavky, které byly na simulátor kladeny. Spolehlivě simuluje daný technologický proces a v reálném čase komunikuje s PLC systémem. Spojení simulátoru s PLC je realizováno dvěma kabely s CANON konektory a pro napájení je použit externí síťový zdroj. Rozměry simulátoru jsou větší, než bylo odhadováno, a to z důvodu množství použité elektroniky. I přes větší rozměry má simulátor povedený design a dá se s ním lehce manipulovat, což byly jedny z hlavních požadavků. Simulátor je složen z levné a dostupné elektroniky, což se pozitivně projevuje na jeho ceně. Celé zařízení je navíc modulární, což je výhodné z hlediska opravitelnosti simulátoru. Při jeho poruše stačí nalézt nefunkční modul a vyměnit ho za nový. Oprava je tedy jednoduchá a velice levná.

Podstatnou část práce tvoří virtuální simulátor a virtuální model simulátoru. Virtuální model reprezentuje 3D model reálného simulátoru. Virtuální simulátor je realizován jako knihovna systému 800xA, která představuje software realizující simulaci technologického procesu a grafické zobrazení simulované soustavy. Virtuální simulátor je tedy možné připojit k reálnému či soft PLC systému a testovat řídicí aplikaci technologického procesu. Na virtuálním simulátoru bylo v době zpřísněných vládních opatření z důvodu virové epidemie úspěšně provedeno několik školení kurzu T315. Tím bylo v praxi ověřeno, že virtuální simulátor je plně zaměnitelný s reálným simulátorem. V rámci práce byl vytvořen virtuální i fyzický model simulátoru. Spojení obou modelů realizuje ABB systém 800xA. Lze tedy říci, že bylo vytvořeno digitální dvojče simulátoru, což je jedna z myšlenek průmyslu 4.0.

Součástí práce je také matlabový skript, který umožňuje graficky zobrazit průběh změny teploty média v nádrži. Přednost skriptu je převážně v jeho obecnosti a tím i široké aplikovatelnosti. Úpravou příslušných konstant lze simulovat průběh teploty libovolné

kapaliny v nádrži o libovolném objemu. Průběh teploty lze graficky zobrazit pro všechny tři případy, ve kterých dochází ke změně teploty kapaliny. Tedy při ohřevu, mísení kapalin i samovolném ochlazení kapaliny skrze stěny nádrže. Výkon topného tělesa lze rovněž v matlabovém skriptu měnit. Práce dále obsahuje obecné řešení diferenciální rovnice druhého řádu, která popisuje změnu teploty při ohřevu. Diferenciální rovnice má koeficienty, které nejsou konstantní. Koeficienty jsou závislé na hmotnosti kapaliny, která se v průběhu ohřevu může měnit. Z tohoto důvodu je nutné rovnici vyřešit pro každý cyklus simulační smyčky. V práci je diferenciální rovnice vyřešena obecně, a to použitím exponenciálních funkcí a kořenů kvadratické rovnice. Tyto matematické operace jsou triviální a není problém obecné řešení implementovat v mikrokontroléru.

Simulace technologického procesu je obecná a simulátor lze parametrizovat pomocí konfigurační aplikace. Aplikace umožňuje připojení osobního počítače k simulátoru a zapsání požadovaných parametrů skrze sériový port. Konfigurace simulátoru je tedy snadná i pro koncového uživatele.

LITERATURA

- [1] *System 800xA Training: T315C Course 04 Addendum Design Specification - RevG*. ABB University.
- [2] *Kalorimetrická rovnice* [online]. [cit. 2021-5-11]. Dostupné z: <http://www.fyzika007.cz/molekulova-fyzika-a-termika/kalorimetricka-rovnice>
- [3] HENNEBERGER, Gerhard. *Electrical Machine I: Basics, Design, Function, Operation*. Aachen University, 2003, s. 66.
- [4] *System 800xA Training: T315C-04 Addendum IO List – RevG*. ABB University.
- [5] *System 800xA Training: T315C-02 System Architecture – RevG*. ABB University.
- [6] *Časopis Elektro* [online]. 6. FCC Public, 2004 [cit. 2021-02-18]. Dostupné z: <http://www.odbornecasopisy.cz/elektro/casopis/tema/ochrana-motoru--14050>
- [7] *Drátek Návodů: Vše o napájení Arduina* [online]. [cit. 2021-02-22]. Dostupné z: <https://navody.drateg.cz/technikuv-blog/napajeni-arduina.html>
- [8] *Arduino: Mega 2560 rev3* [online]. [cit. 2021-02-18]. Dostupné z: <https://store.arduino.cc/arduino-mega-2560-rev3>
- [9] *ICStation: 24V to 5V converter* [online]. [cit. 2021-02-22]. Dostupné z: <http://www.icstation.com/channel-8bit-photoelectric-isolation-module-level-voltage-converter-output-signal-module-p-11592.html>
- [10] *GME: Relé modul 5V, 8x 10A/250V, Low trigger* [online]. [cit. 2021-02-22]. Dostupné z: <https://www.gme.cz/rele-modul-8-kanalu-s-optickym-oddelenim#product-detail>
- [11] *LaskaArduino: Převodník signálu z 4-20mA na V* [online]. [cit. 2021-02-22]. Dostupné z: <https://www.laskarduino.cz/prevodnik-signalu-z-4-20ma-na-v/>
- [12] *ICStation: PWM to Voltage Converter* [online]. [cit. 2021-02-22]. Dostupné z: <http://www.icstation.com/voltage-converter-module-adjustable-converter-power-module-digital-analog-signal-p-12498.html>
- [13] *Drátek: I2C Modulový driver servo motoru* [online]. [cit. 2021-02-22]. Dostupné z: <https://drateg.cz/arduino/1686-iic-i2c-modulovy-driver-servo-motoru-pro-arduino-pca9685-16-kanalu-12-bit-pwm.html>
- [14] *Drátek: PCF8574 I/O Expander* [online]. [cit. 2021-02-22]. Dostupné z: https://drateg.cz/arduino/1380-pcf8574-i-o-expander-i2c-bus-rozsirujici-deska.html?gclid=CjwKCAiAyc2BBhAaEiwA44-wW10B3WOvNRplkLgMF9z3e9I8z7mRk5WzULcTBcoSGvGb0NBhNDMOzxoCDmgQAvD_BwE
- [15] *GME: Modul DC/DC měnič step-down 3-40V/3A* [online]. [cit. 2021-02-22]. Dostupné z: <https://www.gme.cz/napajeci-modul-3-40v-3a-step-down#product-detail>
- [16] *Seed: Grove LED Bar v2.0* [online]. [cit. 2021-02-22]. Dostupné z: <https://www.seeedstudio.com/Grove-LED-Bar-v2-0.html>

- [17] *Ebay: 51seg LED Bargraph Module* [online]. [cit. 2021-02-22]. Dostupné z: <https://www.ebay.com/itm/51seg-LED-Bargraph-Module-DC5V-Isolated-Power-Supply-Emerald-0-5V-05GB7021/153656921699>
- [18] *Drátek: KY-016 RGB LED Modul* [online]. [cit. 2021-02-22]. Dostupné z: <https://dratek.cz/arduino/1403-ky-016-rgb-led-modul-3-barvy-pro-arduino-avr-pic-raspberry.html>
- [19] *Drátek: RGB LED modul 8 x NeoPixel WS2812* [online]. [cit. 2021-02-22]. Dostupné z: <https://dratek.cz/arduino/7718-rgb-led-modul-8-x-neopixel-ws2812.html>
- [20] *GME: LED displej hodinový* [online]. [cit. 2021-02-22]. Dostupné z: <https://www.gme.cz/hodinovy-led-displej>
- [21] *GME: DC ventilátor SUNON MF40100V2-1000U-A99* [online]. [cit. 2021-02-22]. Dostupné z: <https://www.gme.cz/ventilator-sunon-mf40100v2-1000u-a99>
- [22] *GME: Elektromotor 6V RF-300CA* [online]. [cit. 2021-02-22]. Dostupné z: <https://www.gme.cz/elektromotor-rf-300ca-1-5-6v>
- [23] *ICStation: 4 Channel Voltage Level Shifter* [online]. [cit. 2021-02-22]. Dostupné z: <http://www.icstation.com/channel-voltage-level-shifter-translator-optocoupler-isolation-board-signal-4bit-converter-module-p-8042.html>
- [24] *L78 datasheet* [online]. [cit. 2021-02-22]. Dostupné z: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjAudnMmf7uAhVCxIsKHWwmDAoQFjAAegQIBBAD&url=https%3A%2F%2Fwww.st.com%2Fresource%2Fen%2Fdatasheet%2F178m.pdf&usg=AOvVaw1DxUuF_epzFZMSi-Hk6KGd
- [25] PARR, E.A. *Industrial Control Handbook*. Industrial Press, 1998, s. 446. ISBN 0-8311-3085-7.
- [26] JURA, Pavel. *Signály a systémy: Spojité systémy*. 3. Brno: VUT skripta, 2017, s. 4-10.
- [27] *Matematické, fyzikální a chemické tabulky: Pro střední školy*. 3. Prometheus, 1988, s. 107. ISBN 80-85849-84-4.
- [28] *UM10204: I2C-bus specification and user manual* [online]. 6-4 April 2014 [cit. 2021-03-05]. Dostupné z: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- [29] *Arduino.h* [online]. [cit. 2021-5-12]. Dostupné z: <https://github.com/arduino/ArduinoCore-avr/blob/master/cores/arduino/Arduino.h>
- [30] *Wire.h* [online]. [cit. 2021-5-12]. Dostupné z: <https://github.com/esp8266/Arduino/blob/master/libraries/Wire>
- [31] *TM1637Display.h* [online]. [cit. 2021-5-12]. Dostupné z: <https://github.com/avishorp/TM1637>
- [32] *Grove_LED_Bar.h* [online]. [cit. 2021-5-12]. Dostupné z: https://github.com/Seeed-Studio/Grove_LED_Bar

- [33] PCF8574.h [online]. [cit. 2021-5-12]. Dostupné z:
https://github.com/skywodd/pcf8574_arduino_library/blob/master/PCF8574/PCF8574.h
- [34] Adafruit_PWMServoDriver.h [online]. [cit. 2021-5-12]. Dostupné z:
<https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library>
- [35] Adafruit_NeoPixel.h [online]. [cit. 2021-5-12]. Dostupné z:
https://github.com/adafruit/Adafruit_NeoPixel

SEZNAM PŘÍLOH

Příloha 1. – Elektronicky odevzdáno do IS VUT FEKT

Obsahuje:

- Kód řídicí aplikace
- Kód konfigurační aplikace
- Matlab skript pro simulování teploty
- Proteus projekt Arduino emulátoru
- Podrobné schéma zapojení simulátoru
- Uživatelský manuál simulátoru
- Potvrzení o vytvoření zbylé výrobní dokumentace

Příloha 2. – CD, které je součástí tištěné práce

Navíc obsahuje obrázky použité v práci.